

**Construction
Specifications for the
Ticket Research File:
TRF10
Volume 2**

September 21, 2012

*Lesley Hildebrand
Laura Kosar
Xiao Barry
Jeremy Page
Patsy Lloyd
Miriam Loewenberg
Dawn Phelps
Natalie Hazelwood
Svetlana Bronnikov*

MATHEMATICA
Policy Research, Inc.

Contract Number:
SS00-11-31447

Mathematica Reference Number:
06979-160

Submitted to:
Social Security Administration
Office of Disability and Income Support
ITC Building
500 E. Street, SW, 9th Floor
Washington, DC 20254
Project Officer: Paul O'Leary

Submitted by:
Mathematica Policy Research
1100 1st Street NE
Suite 1200
Washington, DC 20024
Telephone: (202) 484-9220
Facsimile: (202) 863-1763
Project Director: Lesley Hildebrand & Laura Kosar

**Construction
Specifications for the
Ticket Research File:
TRF10
Volume 2**

September 21, 2012

Lesley Hildebrand
Laura Kosar
Xiao Barry
Jeremy Page
Pasty Lloyd
Miriam Loewenberg
Dawn Phelps
Natalie Hazelwood
Svetlana Bronnikov

MATHEMATICA
Policy Research, Inc.

Overview of TRF documentation

This document is one of several that provide information about the Ticket Research File (TRF) to help researchers understand and use its data. The full TRF documentation includes the User's Guide, the Data Dictionary, and the Construction Guide. All of these documents are available to SSA staff on the SSA intranet site at <http://ordp.ba.ad.ssa.gov/OPDR/content/research#daf>. Other TRF users can contact OPDR.DAF@ssa.gov with inquiries regarding TRF documentation.

Descriptions of the documents that comprise the TRF10 data documentation

User's Guide for the Ticket Research File: TRF10 Volumes I and II (Hildebrand et al. 2012). The purpose of the User's Guide is to assist users in understanding the TRF10 and related files. Volume I provides an overview of the structure of the TRF and related files, and a summary of key variables are described in more detail in the TRF10 Data Dictionary. It describes methods to use and link TRF files and other data sources for research purposes. Volume II contains detailed appendices, including file layouts for the data sources described in Volume I.

Data Dictionary for the Ticket Research File: TRF10 Volumes I and II (Hildebrand et al. 2012). The Data Dictionary provides detail on TRF variables. Volume I includes an overview of the structure of the various TRF components, a list of variables, and an explanation of each of the variable description fields in Volume II. Volume II contains specifications for each variable, including name, definition, data format, identification of the TRF component to which it belongs, data source, availability, and selected SAS code used to construct the variable. Selected variables are annotated with additional information and context for researchers.

Construction Specifications for the Ticket Research File (*this document*): **TRF10 Volumes I and II** (Hildebrand et al. 2012). The Construction Specifications provide technical details on the construction of TRF10. Volume I describes the construction and validation process, and mainframe data processing methods. The appendices collected in Volume II include the SAS and JCL code used in constructing TRF10, validation tables and charts used, and a glossary of terms.

This page has been left blank for double-sided copying.

CONTENTS

VOLUME I

I	INTRODUCTION	I
	A. Overview of the Construction Guide.....	I.1
	B. Description of the TRF Database.....	I.1
	C. Structure of the TRF Database	I.1
	D. Structure of the Construction Guide.....	I.4
II	TESTING AND VALIDATION	II
	A. Overview & Methods.....	II.1
	B. TRF10 Results	II.2
III	ASSESSING DATA NEEDS FOR TRF10.....	III
	A. Overview of the Process.....	III.1
	B. Process for Assessing Data Needs	III.1
IV	SYSTEMS INFORMATION	IV
	A. Mainframe Data Processing	IV.1
V	CONSTRUCTION STEPS.....	V
	TASK 1: Assemble & Combine DBAD Files.....	V.1
	TASK 2: Assemble & Combine CER 100% Files.....	V.4
	TASK 3: Create Finders	V.7
	TASK 4: Submit Finders	V.10
	TASK 5: Process 831& 832/833 Data.....	V.12
	TASK 6: Process MBR Data: Part A.....	V.17
	TASK 7: Process MBR Data: Part B	V.23
	TASK 8: Process SSI-LF Data	V.28
	TASK 9: Process Numident Data.....	V.32
	TASK 10: Create TRF Demo Data.....	V.36

TASK 11: Create TRF Ticket ComponentV.40
TASK 12: Format Earnings DataV.45
TASK 13: Left Due to Work FlagsV.48
TASK 14: Create TRF Annual Data.....V.53
TASK 15: Create Payments Data.....V.60
TASK 16: Create RSA Linkable File.....V.61

VOLUME II

APPENDIX A: PROGRAMS & JCL LISTINGS

APPENDIX B: TABLES/CHARTS

APPENDIX C: GLOSSARY

APPENDIX A
PROGRAMS & JCL LISTINGS

Appendix A.1

JCL:HDBAJCL

```
// $8043CDB JOB (12510000,T715,,SAS,,ITC9FL),$8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043,
//          RESTART=JS100.SAS9
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(HDBAJCL) |
//*          *-----*
//*
//*          1. ALL STEPS ARE STEP RESTARTABLE
//*          2. SAS PROGRAM CODE MUST BE SAVED IN A LIBRARY
//*              IN A MEMBER NAMED HISTDBAD
//*          READ IN DBAD - ALL HIST 2010
//*          JCL CALLS THE APPROPRIATE PROGRAM
//*          LAYOUT CHANGE IN DEC 2010 - DECEMBER 2009 LAYOUT IS THE SAME
//*          EXCEPT FOR A FEW CHANGES AT THE END - USE LAST INPUT STATEMENT
//*          FORM 2010.
//*          THE DECEMBER CHANGE DOES NOT IMPACT THE VARS WE SELECT
//*          PROGRAMMER MIRIAM LOEWENBERG
//*          SSA PHONE 202 358-6214 MPR 202 484-4829
//*          E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//JS010 EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1 DD DSN=MTOSSI.T2.DBADMBR.D1001,DISP=SHR
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1001.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1 DD DSN=MTOSSI.T2.DBADMBR.D1002,DISP=SHR
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1002.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1 DD DSN=MTOSSI.T2.DBADMBR.D1003,DISP=SHR
//TEMP DD DSN=&&TEMP,
```

Appendix A.1

JCL:HDBAJCL

```
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1003.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS040    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1004,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1004.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS050    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1005,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1005.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS060    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1006,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1006.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS070    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1007,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
```

Appendix A.1

JCL:HDBAJCL

```
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1007.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS080    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1008,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1008.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS090    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1009,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1009.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS100    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1010,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1010.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS110    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.T2.DBADMBR.D1011,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
```

Appendix A.1

JCL:HDBAJCL

```
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1011.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//      DISP=(SHR,PASS,KEEP)
//*
//JS120 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN1 DD DSN=MTOSSI.T2.DBADMBR.D1012,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1012.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTDBAD),
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.2

SAS Code: HISTDBAD

```
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;
/* LAYOUT FOR JUN-DEC 2009 */
/* NEW FORMAT IN DEC BUT OUR FIELDS HAVE NOT MOVED */
DATA TEMP.DBAD;
  INFILE IN1 TRUNCOVER;
  INPUT
  @001 SEL_DATE YYMMDD8.
  @009 CAN    $9.          /* CLAIM ACCOUNT NUMBER (SSN)    */
  @376 BIC    $2.          /* BENE IDENTIFICATION CODE      */
  @378 LAF    $2.          /* LEDGER ACCOUNT FILE          */
  @380 DOB    YYMMDD8.     /* DATE OF BIRTH                */
  @1113 BOAN  $9.          /* BENEFICIARY OWN ACCOUNT NUMBER*/
  @3297 TOC   $1. /* TYPE OF CLAIM */
  @3321 ZIP    $5.          /* ZIP CODE                      */
  @3326 ZIPADD $4.          /* ZIP ADD ON CODE              */
  ;

  /* COMPUTE AGE AT THE TIME OF THE EXTRACT */

  AGE = (SEL_DATE - DOB)/365.25;

  /* COMPUTE RETIREMENT DATE FOR COMPARISON */
  /* IF B'DAY IS JANUARY 1ST - REFER TO PREVIOUS YEAR */
  /* SAS9 OFFERS "SAMEDAY" WHICH SIMPLIFIES THE CODE BELOW */
  /* BASIS FOR CODE IS PUBLISHED TABLE WHICH GIVES RETIREMENT AGE
  ACCORDING TO BIRTH YEAR */

  IF MONTH(DOB) = 1 AND DAY(DOB) = 1 THEN DO;
    YEARDOB = YEAR(DOB) - 1;
    MONTHDOB = 12;
    DAYDOB = 31;
  END;

  ELSE DO;
    YEARDOB = YEAR(DOB);
    MONTHDOB = MONTH(DOB);
    DAYDOB = DAY(DOB);
  END;

  BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
  IF YEARDOB LE 1937 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, (65*12), "SAMEDAY");
  ELSE IF YEARDOB = 1938 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAMEDAY");
  ELSE IF YEARDOB = 1939 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAMEDAY");
  ELSE IF YEARDOB = 1940 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAMEDAY");
  ELSE IF YEARDOB = 1941 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAMEDAY");
  ELSE IF YEARDOB = 1942 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAMEDAY");
  ELSE IF YEARDOB GE 1943 AND YEARDOB LE 1954 THEN RETIREDATE =
```

Appendix A.2

SAS Code: HISTDBAD

```
INTNX('MONTH',BIRTHDATE,(66*12),"SAMEDAY");
ELSE IF YEARDOB = 1955 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,((66*12)+2),"SAMEDAY");
ELSE IF YEARDOB = 1956 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,((66*12)+4),"SAMEDAY");
ELSE IF YEARDOB = 1957 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,((66*12)+6),"SAMEDAY");
ELSE IF YEARDOB = 1958 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,((66*12)+8),"SAMEDAY");
ELSE IF YEARDOB = 1959 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,((66*12)+10),"SAMEDAY");
ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
  INTNX('MONTH',BIRTHDATE,(67*12),"SAMEDAY");

/* COMPUTE AGE AT RETIREMENT */

FRTRAGE = (RETIREDATE-DOB)/365.25;

/* CONDITIONS FOR INCLUSION */

BICC = 'N';
LAFC = 'N';

IF SUBSTR(BIC,1,1) = 'A' AND TOC IN ('5' '6') OR
  SUBSTR(BIC,1,1) = 'C' AND TOC IN ('3' '4' '7') OR
  SUBSTR(BIC,1,1) = 'W' THEN BICC = 'Y';
IF UPCASE(LAF) IN ('AD' 'AS' 'A9' 'C' 'D' 'DP' 'DW' 'E' 'S' 'SD'
'S0' 'S1' 'C2'
'S2' 'S6' 'S8')
  THEN LAFC = 'Y';

IF SUBSTR(BIC,1,1) = 'A' THEN SSN = CAN;
ELSE SSN = BOAN;

IF 18 <= AGE < FRTRAGE AND BICC = 'Y' AND LAFC = 'Y' AND SSN NE ' ';
DROP SEL_DATE LAFC BICC DOB YEARDOB MONTHDOB DAYDOB
  RETIREDATE ;

RUN;
/* RETAIN ALL SSN/BIC COMBINATIONS SO THAT NO CAN'S WILL BE LOST */
PROC SORT DATA=TEMP.DBAD OUT=OUT1.DBAD NODUPKEY ;
  BY SSN BIC;
RUN;
PROC FREQ DATA=OUT1.DBAD; TABLES AGE FRTRAGE BIC;
  FORMAT AGE FRTRAGE 8.1; RUN;
PROC CONTENTS DATA=OUT1.DBAD; RUN;
PROC PRINT DATA=OUT1.DBAD (OBS=50 );
RUN;
```

Appendix A.3

JCL/SAS Code: COMBDBAD

```

//$8043CRM JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(COMBDBAD) |
//*          *-----*
//* COMBINE ALL THE DBAD FILES FOR 2010
//* CASES IN DBAD FROM JAN 2009-DECEMBER 2010
//* PROGRAMMER MIRIAM LOEWENBERG
//* MPR PHONE 202 484-4829 SSA PHONE 202 358-6214
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*
//* SAVE JOB OUTPUT AS:OPDR.TG.PRD.ETTW.N8043.COMBDBAD.OUT10
//*****
//*
// EXEC SAS9,
//       WORK='120000,60000'
//*
//IN1   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1001.SSD,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1002.SSD,DISP=SHR
//IN3   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1003.SSD,DISP=SHR
//IN4   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1004.SSD,DISP=SHR
//IN5   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1005.SSD,DISP=SHR
//IN6   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1006.SSD,DISP=SHR
//IN7   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1007.SSD,DISP=SHR
//IN8   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1008.SSD,DISP=SHR
//IN9   DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1009.SSD,DISP=SHR
//IN10  DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1010.SSD,DISP=SHR
//IN11  DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1011.SSD,DISP=SHR
//IN12  DD DSN=OPDR.TG.PRD.ETTW.N8043.DBAD1012.SSD,DISP=SHR
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBDBAD.D1012.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;

%let begyr=2010;
%let endyr=2010; /* change as needed */
%let endmn=12;   /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;

```

Appendix A.3

JCL/SAS Code: COMBDBAD

```
%if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
%let x&k=%eval(&yr)%eval(&mn);
%let k=%eval(&k+1);
%end;
%end;
%let tot=%eval(&k-1);

data OUT1.DBAD;
merge
%do i=1 %to 12;
    in&i..DBAD (RENAME=(ZIP=ZIP&&X&I ZIPADD=ZIPA&&X&I))
%end;;
by ssn bic;
if ssn gt '000000000';
run;

proc contents;
run;
%mend;

%start;
```

Appendix A.4

JCL: HSORBAT

```
// $8043SHI JOB (12510000,T715,,SAS,,ITC9FL),8043LOEW,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *****
// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.N8043.LIB11(HSORBAT) |
// *          *-----*
// *
// *          1. ALL STEPS ARE STEP RESTARTABLE
// *          2. SAS PROGRAM CODE MUST BE SAVED IN A LIBRARY
// *              IN A MEMBER NAMED HISTSOR
// * READ IN MONTHLY SORD DATA - JCL CALLS THE PROGRAM HISTSOR
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// JS010 EXEC SAS9,
//          WORK='120000,60000'
// *
// IN1 DD DSN=MTOSI.CER100.FIELD.D1001,DISP=SHR
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
// OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1001.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
// *
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSOR),
//          DISP=(SHR,PASS,KEEP)
// *
// JS020 EXEC SAS9,
//          WORK='120000,60000'
// *
// IN1 DD DSN=MTOSI.CER100.FIELD.D1002,DISP=SHR
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
// OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1002.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
// *
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSOR),
//          DISP=(SHR,PASS,KEEP)
// *
// JS030 EXEC SAS9,
//          WORK='120000,60000'
// *
// IN1 DD DSN=MTOSI.CER100.FIELD.D1003,DISP=SHR
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
// TEMP DD DSN=&&TEMP,
```

Appendix A.4

JCL: HSORBAT

```
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1003.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS040    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.CER100.FIELD.D1004,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1004.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS050    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.CER100.FIELD.D1005,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1005.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS060    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.CER100.FIELD.D1006,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1006.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
```

Appendix A.4

JCL: HSORBAT

```
//*
//JS070 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN1   DD DSN=MTOSSI.CER100.FIELD.D1007,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1007.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//      DISP=(SHR,PASS,KEEP)
//*
//JS080 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN1   DD DSN=MTOSSI.CER100.FIELD.D1008,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1008.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//      DISP=(SHR,PASS,KEEP)
//*
//JS090 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN1   DD DSN=MTOSSI.CER100.FIELD.D1009,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1009.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//      DISP=(SHR,PASS,KEEP)
//*
//JS100 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN1   DD DSN=MTOSSI.CER100.FIELD.D1010,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
```

Appendix A.4

JCL: HSORBAT

```
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1010.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS110    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.CER100.FIELD.D1011,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1011.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
//*
//JS120    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSSI.CER100.FIELD.D1012,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PICKZP11,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1012.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(HISTSORD),
//          DISP=(SHR,PASS,KEEP)
```

Appendix A.5

SAS Code: HISTSOR

```

OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;
/* THIS PROGRAM READS SORD IN LIKE HISTORIC REMICS */
/* VARIABLE NAMES ARE SET TO CORRESPOND TO REMICS */

/* CREATE FORMAT STATEMENT FROM UPDATED ZIP-FIPS FILE */

DATA PICKZIP;
  INFILE IN2 TRUNCOVER;
  INPUT @1 RESZIP $5.
        @7 STATE $2.
        @10 CNTY $3. ;
  ZIPCODE = RESZIP;
  FIPSCODE = STATE||CNTY;
RUN;

PROC PRINT DATA=PICKZIP (OBS=25); RUN;
DATA FIPS;
  SET PICKZIP(RENAME=(ZIPCODE=START FIPSCODE=LABEL));
  FMTNAME='$FIPS';
  OUTPUT;
  IF _N_ = 1 THEN DO;
    START='OTHER';
    LABEL = ' ';
  OUTPUT;
  END;
RUN;
PROC FORMAT CNTLIN=FIPS;
RUN;

DATA SORD;
  INFILE IN1 TRUNCOVER;
  INPUT
@0004 MFT          $2.          /* MASTER FILE TYPE */
@0006 HUN          $9.          /* HOUSED UNDER NUMBER */
@0015 PAN          $9.          /* PAN SOCIAL SECURITY NUMBER */
@0108 RCD_EST     ?? YMMDD8.    /* RCD-EST-JD RECORD ESTABLISHMENT
                                DATE */
@0116 BIRTH_JD    ?? YMMDD8.    /* BIRTH-JD DATE OF BIRTH */
@0134 SEX         $1.          /* SEX SEX */
@0135 RACE        $1.          /* RACE */
@204  DENCDE      $3.          /* DENIAL CODE */
@0326 DX          $4.          /* DIB-DIG PRIMARY DISABILITY
                                DIAG. CODE */
@0343 PDS2        $2.          /* PDS2 RESIDENCE STATE CODE */
@0345 PDS3        $3.          /* PDS3 RESIDENCE COUNTY CODE */
@0349 PDZIP1_5    $5.          /* PDZIP1-5 PAYEE'S ZIP CODE */
@0354 PDZIP6_9    $4.          /* PDZIP6-9 PAYEE'S ZIP CODE SUFFIX
*/
@0361 IET4        $1.          /* EARNED INC. TYPE
                                (SELF EMPLOYMENT) */
@0368 IEA4        6.2          /* EARNED INC. AMT.
                                SELF EMPLOYMENT */

```

Appendix A.5

SAS Code: HISTSORT

```

@0375 IET6          $1.          /* EARNED INC. TYPE (WAGES)*/
@0382 IEA6          6.2          /* EARNED INC. AMT. (WAGES) */
@0389 IET1          $1.          /* EARNED INC. TYPE (BLIND W E)*/
@0396 IEA1          6.2          /* EARNED INC. AMT. (BLIND W E)*/
@0403 IET2          $1.          /* EARNED INC. TYPE
(PASS) */
@0410 IEA2          6.2          /* EARNED INC. AMT.
(PASS)*/
@0417 IET5          $1.          /* EARNED INC. TYPE (IRWE)*/
@0424 IEA5          6.2          /* EARNED INC. AMT. (IRWE)*/
@0431 IET3          $1.          /* EARNED INC. TYPE (NET LOSS)*/
@0438 IEA3          6.2          /* EARNED INC. AMT. (NET LOSS)*/
@;
POS = 445;
ARRAY UT (6) $ T1-T6;
ARRAY UA (6) UN1-UN6;
DO I = 1 TO 6;
  INPUT
  @POS UT(I) $2.
  @POS+8 UA(I) 6.2
  @;
  POS = POS + 27;
END;
INPUT
@0691 AAZIP1_5      $5.          /* AAZIP1-5 APPLICANT'S ADDRESS ZIP
CODE */
@0696 AAZIP6_9      $4.          /* AAZIP6-9 APPLICANT'S ZIP CODE
SUFFIX */
@0794 CPST          $3.          /* PSTAT-CURR - PAYMENT STATUS
CODE*/
@0811 MEDC          $1.          /* MEDIC MEDICAID ELIGIBILITY
CODE */
@0814 EICM          4.           /* EINCM-CURR CHARGEABLE EARNED
INC. AMT. */
@0818 UINC          4.           /* UINCM-CURR CHARGEABLE UNEARNED
INC. AMT. */
@0835 MEDT          $1.          /* MEDTEST-CURR MEDICAL &
SOC. SERV. INC TEST */
@0853 SEL_DATE     ?? YMMDD8.    /* DATE OF SELECTION */
@0990 FL           $2.          /* LANG-PREF-WRITTEN LANGUAGE
CODE */
@1003 CIT           $1.          /* ALIEN ALIEN REFUGE INDICATOR */
;

DO I = 1 TO 6;
  IF UT(I) = 'A' THEN IUA1 = UA(I); /* TYPE II */
  ELSE IF UT(I) = 'F' THEN IUA6 = UA(I); /* TANF */
  ELSE IF UT(I) = 'Q' THEN IUA3 = UA(I); /* WORKERS COMP */
END;
DROP UN1-UN6 T1-T6;
IF SUBSTR(MFT,1,1) = 'X' THEN DELETE;
/* NORA'S CONVERSATION WITH M. BARBOUR */

```

Appendix A.5

SAS Code: HISTSORT

```
/* CREATE AGE AT TIME OF EXTRACTION VARIABLE */

AGE = (SEL_DATE - BIRTH_JD)/365.25;

/* CREATE RETIREMENT AGE FOR CUTOFF REFERENCE - NEW 2007 */
/* BASED ON A PUBLISHED TABLE WHICH GIVES RETIREMENT AGE ACCORDING
TO
    BIRTH YEAR */

/* IF B'DAY IS JANUARY 1ST - REFER TO PREVIOUS YEAR */
IF MONTH(BIRTH_JD) = 1 AND DAY(BIRTH_JD) = 1 THEN DO;
    YEARDOB = YEAR(BIRTH_JD) - 1;
    MONTHDOB = 12;
    DAYDOB = 31;
END;

ELSE DO;
    YEARDOB = YEAR(BIRTH_JD);
    MONTHDOB = MONTH(BIRTH_JD);
    DAYDOB = DAY(BIRTH_JD);
END;
IF MONTHDOB = 2 AND DAYDOB GT 28 THEN DAYDOB = 28;

    BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
    IF YEARDOB LE 1937 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, (65*12), "SAME DAY");
    ELSE IF YEARDOB = 1938 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAME DAY");
    ELSE IF YEARDOB = 1939 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAME DAY");
    ELSE IF YEARDOB = 1940 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAME DAY");
    ELSE IF YEARDOB = 1941 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAME DAY");
    ELSE IF YEARDOB = 1942 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAME DAY");
    ELSE IF YEARDOB GE 1943 AND YEARDOB LE 1954 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, (66*12), "SAME DAY");
    ELSE IF YEARDOB = 1955 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((66*12)+2), "SAME DAY");
    ELSE IF YEARDOB = 1956 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((66*12)+4), "SAME DAY");
    ELSE IF YEARDOB = 1957 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((66*12)+6), "SAME DAY");
    ELSE IF YEARDOB = 1958 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((66*12)+8), "SAME DAY");
    ELSE IF YEARDOB = 1959 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, ((66*12)+10), "SAME DAY");
    ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
        INTNX('MONTH', BIRTHDATE, (67*12), "SAME DAY");

/* COMPUTE AGE AT RETIREMENT */
```

Appendix A.5

SAS Code: HISTSORT

```
FRTRAGE = (RETIREDATE-BIRTH_JD)/365.25;

IF 10 <= AGE < FRTRAGE; /* REVISED 2006 TO CAPTURE YTD KIDS */

INCLUDE = 'N';

IF UPCASE(MFT) IN('DI' 'BI' 'DC' 'BC' 'DS' 'BS') THEN
  INCLUDE = 'Y';

IF AAZIP1_5 > '00000' THEN DO;
  RESZIP = AAZIP1_5;
  RZIPS = AAZIP6_9;
END;
ELSE IF AAZIP1_5 <= '00000' THEN DO;
  RESZIP = PDZIP1_5;
  RZIPS = PDZIP6_9;
END;

/* REVISION 2010 WITH LATER LIST OF ZIP CODES */

FIPS = PUT(RESZIP,$FIPS.);
LENGTH ST $2 CNTY $3;
ST=SUBSTR(FIPS,1,2);
CNTY=SUBSTR(FIPS,3,3);
ACT='N';

IF UPCASE(CPST) IN ('C01' 'E01' 'E02' 'M01' 'M02'
                  'P01' 'S05' 'S06' 'S07' 'S08' 'S09' 'S21' 'T30'
                  'T32' 'T33') THEN ACT = 'Y';

IF DENCDE = ' ' THEN DO; /* NEW IN 2004 TO ELIM DENIALS FOR N CODES
*/
  IF UPCASE(CPST) IN ('N01' 'N02' 'N04'
                    'N05' 'N06' 'N09' 'N20' 'N22' 'N33' 'N44' 'N54')
    THEN ACT = 'Y';
  END;

  IF ACT = 'Y' AND INCLUDE = 'Y';

LENGTH BIRTH_JD RCD_EST 4
EICM UINC IEA1 IEA2 IEA3 IEA4 IEA5 IEA6 IUA1 IUA6 IUA3 4;

FORMAT EICM UINC 7.2
IEA1 IEA2 IEA3 IEA4 IEA5 IEA6 IUA1 IUA6 IUA3 11.2
BIRTH_JD RCD_EST YYMMDD10.;
;

DROP SEL_DATE YEARDOB MONTHDOB DAYDOB
INCLUDE ACT ;

IF PAN GT '000000000';

RUN;
```

Appendix A.5
SAS Code: HISTSORT

```
PROC SORT DATA=SORD OUT=TEMP.SORD;  
  BY PAN RCD_EST;  
RUN;
```

```
DATA OUT1.SORD;  
  SET TEMP.SORD;  
  BY PAN;  
  IF LAST.PAN;  
RUN;
```

```
PROC FREQ DATA=OUT1.SORD; TABLES AGE FRTRAGE MFT CPST;  
  FORMAT AGE FRTRAGE 8.1; RUN;  
PROC CONTENTS DATA=OUT1.SORD; RUN;  
PROC PRINT DATA=OUT1.SORD (OBS=25); RUN;
```

Appendix A.6

JCL/SAS Code: COMBSORD

```

//$8043CRM JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB10(COMBSORD) |
//*          *-----*
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE: 202 358-6214 MPR PHONE: 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* COMBINE THE MONTHLY SORD EXTRACTS - OUTPUT 1 RECORD PER BENE
//*****
//*
// SET REG='64M'
//JS010 EXEC SAS9,
//          WORK='200000,100000',          * SPACE IN BLKS *
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1001.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1002.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1003.SSD,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1004.SSD,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1005.SSD,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1006.SSD,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1007.SSD,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1008.SSD,DISP=SHR
//IN9 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1009.SSD,DISP=SHR
//IN10 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1010.SSD,DISP=SHR
//IN11 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1011.SSD,DISP=SHR
//IN12 DD DSN=OPDR.TG.PRD.ETTW.N8043.SORD1012.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBSORD.D1012.SSD1,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD *

```

```

OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;

```

```

%let begyr=2010;
%let endyr=2010; /* change as needed */
%let endmn=12;   /* change as needed */

```

```

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%EVAL(&j);

```

Appendix A.6
JCL/SAS Code: COMBSORD

```
        %else %let mn=%eval(&j);
        %let x&k=%eval(&yr)%eval(&mn);
        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

data out1.SORD;
  merge
  %do i=1 %to &TOT;
    in&i..sord(rename=(
  hun=  hun&&x&i
  cit=  cit&&x&i
  dx=   dx&&x&i
  eicm= eicm&&x&i
  fl=   fl&&x&i
  iet1= iet1&&x&i
  iet2= iet2&&x&i
  iet3= iet3&&x&i
  iet4= iet4&&x&i
  iet5= iet5&&x&i
  iet6= iet6&&x&i
  ieal= ieal&&x&i
  ieal2= ieal2&&x&i
  ieal3= ieal3&&x&i
  ieal4= ieal4&&x&i
  ieal5= ieal5&&x&i
  ieal6= ieal6&&x&i
  iual= iual&&x&i
  iual6= iual6&&x&i
  iual3= iual3&&x&i
  medc= medc&&x&i
  medt= medt&&x&i
  pds2= pds2&&x&i
  pds3= pds3&&x&i
  cpst= cpst&&x&i
  uinc= uinc&&x&i
  reszip = rzip&&x&i
  rzips = rzps&&x&i
  st=    st&&x&i
  cnty=  cnty&&x&i)
  )
  %end;;
  by pan;
  IF PAN GT '000000000';

  /* THE MACRO VARIABLES ALLOW THE CREATION OF YY/MM VARIABLES USING
     THE EXTRACTS READ IN FROM THE MONTHLY SORD FILES */
run;

proc contents;
PROC PRINT DATA=OUT1.SORD (OBS=25); RUN;
run;
```

Appendix A.6
JCL/SAS Code: COMBSORD

%mend;

%start;

Appendix A.7
JCL/SAS Code: MAKEFIND

```
// $8043FIN JOB (12510000,T715,,SAS,,ITC9FL),8043LOWE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=N8043
// SET REG='64M'
//JOB01 EXEC SAS9,
//          WORK='180000,90000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(MAKEFIND) |
//*          *-----*
//* DEVELOP FILE OF SSN'S FOR ALL IN FINDERS TRF1-2-05-06-07-08-09-
//*                                     10
//* CREATE MASTER FILE OF ALL SSN'S TRF2-TRF10
//* FOR SUBMITTING TO NUMIDENT AND EARNINGS DATA
//* CREATE LINKING FILE FOR T2 RETURN RECORDS LINK
//* USE 2010 FINDER FILES + NEW CASES - 2010 HAD ALL CASES IN DEMO
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6229 MPR PHONE 202 264-3447
//*****
//*
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//TEMP2 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//TEMP3 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//TEMP4 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.COMDBBAD.D1012.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBSORD.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D0912.SSD,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D0912.FLAT,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T2FIND.D1012.FLAT,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),
//          DCB=(RECFM=FB,LRECL=80,DSORG=PS)
//OUT2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),
//          DCB=(RECFM=FB,LRECL=80,DSORG=PS)
//OUT3 DD DSN=OPDR.TG.PRD.ETTW.N8043.MASTFIND.D1012.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//OUT4 DD DSN=OPDR.TG.PRD.ETTW.N8043.MASTFIND.D1012.FLAT,
```

Appendix A.7

JCL/SAS Code: MAKEFIND

```
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),
//                                     DCB=(RECFM=FB,LRECL=80,DSORG=PS)
*****
/*****
* FILENAME: MAKEFIND
* PROGRAMMER:MIRIAM LOEWENBERG
* PURPOSE:CREATE FINDER AND LINKING FILES FOR SSI-DI TRF10
* MAKE SURE WE HAVE ALL THE SSN/BIC COMBOS IN DI LINKING FILE
* CREATED:01/21/09
* UPDATED: 02/08/12 FOR TRF10
*****/
options nocenter ls=132 ps=60 compress=YES;
/* LOAD THE 2009 FINDER FILES AND ADD ON 2010 RECORDS */

DATA SSIFIND;
  INFILE IN4 TRUNCOVER;
  INPUT @001 SSN $9.
  ;
RUN;

/* CREATE TEMP FILE FROM COMBINED 2010 DBAD FILES */
DATA TEMP2.DBAD (KEEP=SSN CAN BIC);
  SET IN1.DBAD;
  BY SSN BIC;
RUN;

/* CREATE TEMP FILE FROM COMBINED 2010 SORD FILES */
DATA SORD (KEEP=SSN);
  SET IN2.SORD;
  SSN = PAN;
RUN;

/* COMBINE NEW DI LINKING FILE WITH OLD DI LINK */
/* OUTPUT FILE WILL BE THE LINK FOR KEEPING MBR RECORDS
   RETURNED FROM THE FINDER PROCESS */

DATA OUT.LINKDI;
  MERGE IN3.LINKDI TEMP2.DBAD;
  BY SSN BIC;
  /* GET RID OF BLANK CAN'S */
  IF CAN NE ' ';
RUN;

PROC PRINT DATA=OUT.LINKDI (OBS=25);TITLE 'LINKDI'; RUN;

/* CREATE DI FINDER OF CAN'S */
PROC SORT DATA=OUT.LINKDI NODUPKEY OUT=TEMP3.T2FIND; BY CAN; RUN;

/* RESORT T2 DATA DE-DUPING ON SSN FOR MASTER FILE */
PROC SORT DATA=OUT.LINKDI NODUPKEY OUT=TEMP4.T2FIND; BY SSN; RUN;
```

Appendix A.7
JCL/SAS Code: MAKEFIND

```
/* CREATE THE T16 FINDER BY COMBINING OLD AND NEW */

DATA TEMP1.T16FIND;
  MERGE SSIFIND SORD; BY SSN;
RUN;

PROC PRINT DATA=TEMP3.T2FIND (OBS=25); TITLE 'T2FIND'; RUN;
PROC PRINT DATA=TEMP1.T16FIND (OBS=25);TITLE 'T16FIND'; RUN;

/* OUTPUT FLAT FINDER FILES FOR PROCESSING BY SSA STAFF*/

DATA NULL;
  SET TEMP3.T2FIND;
  BY CAN;
  file out1;
  put
  @1 CAN $9.
  ;
RUN;
DATA NULL;
  SET TEMP1.T16FIND;
  BY SSN;
  file out2;
  put
  @1 SSN $9.
  ;
RUN;

/* OUTPUT MASTER FIND FOR NUMIDENT AND EARNINGS DATA */
/* OUTPUT MASTER FLAT DATA AND SAS DATA SET */
/* FLAT FILE IS SENT TO SSA STAFF AS FINDER - SAS DATA SET IS
   USED TO SCREEN THE RETURNED RECORDS */

DATA OUT3.MASTFIND;
  MERGE TEMP1.T16FIND TEMP4.T2FIND;
  BY SSN;
  file out4;
  put
  @1 SSN $9.
  ;
RUN;
PROC PRINT DATA=OUT3.MASTFIND (OBS=25);TITLE 'MASTFIND'; RUN;
```

Appendix A.8

JCL/SAS Code: LINKFIND

```

//$2358LNK JOB (11710000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.$2358.TRF10.PRDLIB(LINKFIND) |
//*          *-----*
//* CONTACT NATALIE HAZELWOOD
//* SSA PHONE: 202 358-6228 MPR PHONE: 202 264-3447
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
//* CREATE LINK FOR SUBSETTING 831-33 RECORDS
//*****
//*
// SET REG='64M'
//JS010 EXEC SAS9,
//          WORK='100000,50000',          * SPACE IN BLKS *
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T2FIND.D1012.FLAT,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,50),RLSE)
//SYSIN DD *
OPTIONS NOCENTER LS=132 PS=60 ;

/* T2FIND CONTAINS CANS FROM THE DBAD AND DEMO DATA DE-DUPLICATED ON
CAN. IN THE LINKDI FILE THE BIC DISTINGUISHES BETWEEN THEM.
T16FIND CONTAINS SSNS FROM NEW SORD AND DEMO .
THE "FIND" FILES ARE FLAT FILES SENT TO SSA AS FINDERS.
831-33 FILE HAS THE DUPLICATED CAN FOR MBR AND THE PAN FOR SSR.
TO LINK BACK WE NEED TO MATCH THE CAN AND THE BIC FOR MBR RECORDS
AND THE PAN FOR SSR. ALL SSR RECORDS HAVE BIC = A IN 831.
CHILDREN ARE REGISTERED AS PRIMARIES ON SSR-831.
THE LINK SHOULD FIND ALL THE RECORDS WE NEED. */

DATA T2;
  INFILE IN1 TRUNCOVER;
  INPUT @01 SSN $9.;/* CALL THE FIELD SSN - IT IS ACTUALLY CAN */
RUN;
DATA T16;
  INFILE IN2 TRUNCOVER;
  INPUT @01 SSN $9.;
RUN;
DATA LINKFIND;
  SET T2 T16; BY SSN;
RUN;
PROC SORT NODUPKEY OUT=OUT.LINKFIND; BY SSN; RUN;

```

Appendix A.9
JCL: JCL31331

```
// $2358R31 JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*
//* COMMENT OUT LINE BELOW
//* RESTART=JS030.SAS9
//*****
//*
// *-----*
// | CODE ORIGINALLY WRITTEN IN: |
// | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(JCL3133A) |
// | |
// | THIS JCL RUNS YEARS 1988-1989 |
// *-----*
//
// 1. ALL STEPS ARE STEP RESTARTABLE
// 2. SAS PROGRAM CODE MUST BE FROM A LIBRARY
// IN MEMBERS NAMED PROG831 AND PROG833
// 3. FOR 2007 PROGRAMS RUN BY MIRIAM - USING DISK VERSIONS
// AFTER CONSULTING WITH JOAN BURKE
// JCL CALLS THE APPROPRIATE PROGRAM FOR READING 831 OR 833 DATA
// CONTACT NATALIE HAZELWOOD
// SSA PHONE 202.358.6228 MPR PHONE 202 264-3447
// E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
// READ IN THE 831-833 FILES
// SECOND READ TO CAPTURE THE MISSING CAN'S
//*****
//
//JS010 EXEC SAS9,
// WORK='120000,60000'
//
//IN DD DSN=AIS.F5750DDB.UNI831.CY1988,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1988.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//
//JS020 EXEC SAS9,
// WORK='120000,60000'
//
//IN DD DSN=AIS.F5750DDB.UNI833.CY1988,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1988.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
```

Appendix A.9
JCL: JCL31331

```
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS030    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1989 ,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD ,DISP=SHR
//TEMP    DD DSN=&&TEMP ,
//          DISP=(NEW ,DELETE ,DELETE) ,
//          SPACE=(CYL ,( 500 ,10) ,RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1989.SSD ,
//          DISP=(NEW ,CATLG ,DELETE) ,
//          SPACE=(CYL ,( 500 ,10) ,RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831) ,
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS030    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1989 ,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD ,DISP=SHR
//TEMP    DD DSN=&&TEMP ,
//          DISP=(NEW ,DELETE ,DELETE) ,
//          SPACE=(CYL ,( 500 ,10) ,RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1989.SSD ,
//          DISP=(NEW ,CATLG ,DELETE) ,
//          SPACE=(CYL ,( 500 ,10) ,RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833) ,
//          DISP=( SHR ,PASS ,KEEP )
//*
```

Appendix A.10

JCL: JCL3133A

```

//$2358R31 JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*
//* COMMENT OUT LINE BELOW
//*          RESTART=JS030.SAS9
//*****
//*
//*          *-----*
//*          | CODE ORIGINALLY WRITTEN IN:          |
//*          | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(JCL3133A) |
//*          | THIS JCL RUNS YEARS 1990-1997          |
//*          *-----*
//*
//*          1. ALL STEPS ARE STEP RESTARTABLE
//*          2. SAS PROGRAM CODE MUST BE FROM A LIBRARY
//*             IN MEMBERS NAMED PROG831 AND PROG833
//*          3. FOR 2007 PROGRAMS RUN BY MIRIAM - USING DISK VERSIONS
//* AFTER CONSULTING WITH JOAN BURKE
//* JCL CALLS THE APPROPRIATE PROGRAM FOR READING 831 OR 833 DATA
//* CONTACT NATALIE HAZELWOOD
//* SSA PHONE 202.358.6228      MPR PHONE 202 264-3447
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
//* READ IN THE 831-833 FILES
//* SECOND READ TO CAPTURE THE MISSING CAN'S
//*****
//*
//JS010 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY1990,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1990.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//      DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//      WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY1990,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1990.SSD,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),

```

Appendix A.10
JCL: JCL3133A

```
//          DISP=(SHR,PASS,KEEP)
//*
//JS030    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1991,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1991.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS040    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1991,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1991.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS050    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1992,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1992.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS060    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1992,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1992.SSD,
```

Appendix A.10
JCL: JCL3133A

```
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS070    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1993,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1993.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS080    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1993,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1993.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS090    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1994,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1994.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS100    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1994,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
```

Appendix A.10
JCL: JCL3133A

```
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1994.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS110    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1995,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1995.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS120    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY1995,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1995.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS130    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY1996,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1996.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS140    EXEC SAS9,
//          WORK='120000,60000'
//*
```

Appendix A.10
JCL: JCL3133A

```
//IN      DD DSN=AIS.F5750DDB.UNI833.CY1996,DISP=SHR
//IN2    DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP   DD DSN=&&TEMP,
//        DISP=(NEW,DELETE,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//OUT    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1996.SSD,
//        DISP=(NEW,CATLG,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//SYSIN  DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//        DISP=(SHR,PASS,KEEP)
//*
//JS150  EXEC SAS9,
//        WORK='120000,60000'
//*
//IN      DD DSN=AIS.F5750DDB.UNI831.CY1997,DISP=SHR
//IN2    DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP   DD DSN=&&TEMP,
//        DISP=(NEW,DELETE,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//OUT    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1997.SSD,
//        DISP=(NEW,CATLG,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//SYSIN  DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//        DISP=(SHR,PASS,KEEP)
//*
//JS160  EXEC SAS9,
//        WORK='120000,60000'
//*
//IN      DD DSN=AIS.F5750DDB.UNI833.CY1997,DISP=SHR
//IN2    DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP   DD DSN=&&TEMP,
//        DISP=(NEW,DELETE,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//OUT    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1997.SSD,
//        DISP=(NEW,CATLG,DELETE),
//        SPACE=(CYL,(500,10),RLSE)
//SYSIN  DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//        DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.11
JCL: JCL3133B

```

//$2358R31 JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*
//* COMMENT OUT LINE BELOW
//* RESTART=JS030.SAS9
//*****
//*
//* *-----*
//* | CODE ORIGINALLY WRITTEN IN: |
//* | OPDR.TG.PRD.ETTW.$2358.TRF10.P832.PRDLIB(JCL3133B) |
//* | THIS JCL RUNS YEARS 1998-2005 |
//* *-----*
//*
//* 1. ALL STEPS ARE STEP RESTARTABLE
//* 2. SAS PROGRAM CODE MUST BE FROM A LIBRARY
//* IN MEMBERS NAMED PROG831 AND PROG833
//* 3. FOR 2007 PROGRAMS ALL IS NOW READY S
//* JCL CALLS APPROPRIATE PROGRAM FOR READING 831 OR 833 DATA
//* CONTACT NATALIE HAZELWOOD
//* SSA PHONE 202.358.6228 MPR PHONE 202 264-3447
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
//* READ IN THE 831-833 FILES
//*****
//*
//JS010 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY1998,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1998.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY1998,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1998.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*

```

Appendix A.11
JCL: JCL3133B

```
//JS030 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI831.CY1999,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y1999.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS040 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI833.CY1999,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y1999.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS050 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI831.CY2000,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2000.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS060 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI833.CY2000,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2000.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)
```

Appendix A.11
JCL: JCL3133B

```
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
//JS070 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY2001,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2001.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS080 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2001,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2001.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
//JS090 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY2002,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2002.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS100 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2002,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
```

Appendix A.11
JCL: JCL3133B

```
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2002.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS110    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY2003,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2003.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS120    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY2003,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2003.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
//          DISP=(SHR,PASS,KEEP)
//*
//JS130    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI831.CY2004,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//OUT     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2004.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(500,10),RLSE)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
//          DISP=(SHR,PASS,KEEP)
//*
//JS140    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN       DD DSN=AIS.F5750DDB.UNI833.CY2004,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
```

Appendix A.11
JCL: JCL3133B

```
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2004.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
//JS150 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY2005,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2005.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS160 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2005,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2005.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.12
JCL: JCL3133C

```
// $2358R31 JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*
//*****
//*
//* *-----*
//* | CODE ORIGINALLY WRITTEN IN: |
//* | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(JCL3133C) |
//* | |
//* | THIS JCL RUNS YEARS 2006-2010 |
//* *-----*
//*
//* 1. ALL STEPS ARE STEP RESTARTABLE
//* 2. SAS PROGRAM CODE MUST BE FROM A LIBRARY
//* IN MEMBERS NAMED PROG831 AND PROG833
//* 3. FOR 2006 2007 2008 2009 2010
//* AFTER CONSULTING WITH JOAN BURKE
//* CONTACT NATALIE HAZELWOOD
//* SSA PHONE 202.358.6228 MPR PHONE 202 264-3447
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
//* READ IN THE 831-833 FILES
//*****
//JS010 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY2006,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2006.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2006,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2006.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
// WORK='120000,60000'
//*
```

Appendix A.12
JCL: JCL3133C

```
//IN DD DSN=AIS.F5750DDB.UNI831.CY2007,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2007.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS040 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2007,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2007.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
//JS050 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI831.CY2008,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2008.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),
// DISP=(SHR,PASS,KEEP)
//*
//JS060 EXEC SAS9,
// WORK='120000,60000'
//*
//IN DD DSN=AIS.F5750DDB.UNI833.CY2008,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2008.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(500,10),RLSE)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),
// DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.12
JCL: JCL3133C

```
//JS070 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI831.CY2009,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2009.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS080 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI833.CY2009,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2009.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD833),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS090 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI831.CY2010,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2010.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS100 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN    DD DSN=AIS.F5750DDB.UNI833.CY2010,DISP=SHR  
//IN2   DD DSN=OPDR.TG.PRD.ETTW.$2358.LINKFIND.D1012.SSD,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)  
//OUT   DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2010.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      SPACE=(CYL,(500,10),RLSE)
```

Appendix A.12
JCL: JCL3133C

```
//SYSIN DD DSN=OPDR.TG.PRD.ETW.$2358.TRF10.P831.PRDLIB(RD833),  
// DISP=( SHR,PASS,KEEP)
```

Appendix A.13

SAS Code: RD331

```

/*****
* FILENAME:  OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(RD831)
* PROGRAMMER: KATE BARTKUS
* PURPOSE: TO READ IN THE 831 TEXT FILES
* EDITED: SEPTEMBER 28, 2001 BY N. KHAN
* EDITED: MIRIAM LOEWENBERG 2003, 2004
* EDITED: LESLEY HILDEBRAND, MARCH 2006, ADD DPM
* EDITED: MIRIAM LOEWENBERG, FEB 2007, ADD RB
* EDITED: NATALIE HAZELWOOD, OCT 2011 FOR TRF10
*****/
options nocenter ls=132 ps=60;
data wkd831;
infile in truncover;
input
@1  SSN      $char9.
@10 bic      $char2.
@12 AL       $char1.
@13 rid      $char1.
@22 toc      $char1.
@23 dodec    yymmdd8.
@23 dodecM   yymmN6.
@31 rdt      $char1.
@37 rb       $char2.
@39 dob      yymmdd8.
@54 rpdX     $char4.
@62 rsdx     $char4.
@73 dr       $char1.
@123 ddpb    $char8.
@123 ddpbX   $char6.
@140 ed      $char2.
@165 PD      $char1.
@168 PDD     $char6.
@157 dotX    $char6.
@229 dpm     $char1.
;

/* MONTH MAY BE MISSING ON DISABILITY DATE FIELDS
   IF SO FILL IN WITH JUNE */
IF SUBSTR(DDPBX,5,2) = '88' THEN SUBSTR(DDPBX,5,2) = '06';
IF SUBSTR(DOTX,5,2) = '88' THEN SUBSTR(DOTX,5,2) = '06';
IF DDPBX NE '999999' THEN DDPBM = INPUT(DDPBX,YYMMN6.);
IF DOTX NE '999999' THEN DOT = INPUT(DOTX,YYMMN6.);
decage=floor((dodec-dob)/365.25);
DROP DDPBX DOTX;
run;

/* SUBSET RECORDS TO CASES IN FINDER */
DATA TEMP.WKD831;
MERGE WKD831 (IN=W) IN2.LINKFIND (IN=LNK); BY SSN;
IF W AND LNK;
RUN;
proc sort data=TEMP.wkd831 out=out.WKD831;
by SSN BIC dodec;
run;

```

Appendix A.13
SAS Code: RD331

```
proc contents;  
PROC PRINT DATA=OUT.WKD831 (OBS=25); RUN;  
run;
```

Appendix A.14
SAS Code: RD833

```
/* *****  
* FILENAME: OPDR.TG.PRD.ETTW.N4743.AB2.LIB(RD833)  
* PROGRAMMER: N. KHAN  
* PURPOSE: TO READ IN THE 833 TEXT FILES  
* EDITED: SEPTEMBER, 2003  
* EDITED: MIRIAM LOEWENBERG, 2004 AND 2005  
* EDITED: LESLEY HILDEBRAND, MARCH 2006 TO ADD DPM  
* *****/  
options nocenter ls=132 ps=60;  
data wkd833;  
infile in trunccover;  
input  
@1 SSN $char9.  
@10 bic $char2.  
@12 AL $char1.  
@13 rid $char1.  
@14 toc $char1.  
@15 dodec yymmdd8.  
@15 dodecM yymmN6.  
@23 rdt $char1.  
@29 rfc $char2.  
@32 wrm $char2.  
@34 dob yymmdd8.  
@49 rpx $char4.  
@57 rsdx $char4.  
@68 dr $char1.  
@118 ddpb $char8.  
@118 ddpbx $char6.  
@135 ed $char2.  
@152 dotX $char6.  
@260 cdt $char2.  
@262 DPM $char1.  
;  
/* MONTH MAY BE MISSING ON DISABILITY DATE FIELDS  
IF SO FILL IN WITH JUNE */  
IF SUBSTR(DDPBX,5,2) = '88' THEN SUBSTR(DDPBX,5,2) = '06';  
IF SUBSTR(DOTX,5,2) = '88' THEN SUBSTR(DOTX,5,2) = '06';  
IF DDPBX NE '999999' THEN DDPBM = INPUT(DDPBX,YYMMN6.);  
IF DOTX NE '999999' THEN DOT = INPUT(DOTX,YYMMN6.);  
decage = floor((dodec-dob)/365.25);  
DROP DDPBX DOTX;  
run;  
  
/* SUBSET RECORDS TO CASES IN FINDER */  
DATA TEMP.WKD833;  
MERGE WKD833 (IN=W) IN2.LINKFIND (IN=LNK); BY SSN;  
IF W AND LNK;  
RUN;  
proc sort data=TEMP.wkd833 out=out.WKD833;  
BY SSN BIC DODEC; RUN;  
PROC PRINT DATA=OUT.WKD833 (OBS=25); RUN;  
  
proc contents;
```

Appendix A.14
SAS Code: RD833

```
run;  
PROC PRINT DATA=OUT.WKD833 (OBS=25); RUN;
```


Appendix A.15
JCL/SAS Code: CMB3133

```
//IN28 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2001.SSD,DISP=SHR
//IN29 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2002.SSD,DISP=SHR
//IN30 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2002.SSD,DISP=SHR
//IN31 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2003.SSD,DISP=SHR
//IN32 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2003.SSD,DISP=SHR
//IN33 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2004.SSD,DISP=SHR
//IN34 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2004.SSD,DISP=SHR
//IN35 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2005.SSD,DISP=SHR
//IN36 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2005.SSD,DISP=SHR
//IN37 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2006.SSD,DISP=SHR
//IN38 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2006.SSD,DISP=SHR
//IN39 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2007.SSD,DISP=SHR
//IN40 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2007.SSD,DISP=SHR
//IN41 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2008.SSD,DISP=SHR
//IN42 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2008.SSD,DISP=SHR
//IN43 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2009.SSD,DISP=SHR
//IN44 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2009.SSD,DISP=SHR
//IN45 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.Y2010.SSD,DISP=SHR
//IN46 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P833.Y2010.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.$2358.CMB3133.Y2010.SSD,
// DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=YES;
DATA OUT1.COMB3133;
  SET
  IN1.WKD831
  IN2.WKD833
  IN3.WKD831
  IN4.WKD833
  IN5.WKD831
  IN6.WKD833
  IN7.WKD831
  IN8.WKD833
  IN9.WKD831
  IN10.WKD833
  IN11.WKD831
  IN12.WKD833
  IN13.WKD831
  IN14.WKD833
  IN15.WKD831
  IN16.WKD833
  IN17.WKD831
  IN18.WKD833
  IN19.WKD831
  IN20.WKD833
  IN21.WKD831
  IN22.WKD833
  IN23.WKD831
  IN24.WKD833
  IN25.WKD831
  IN26.WKD833
  IN27.WKD831
```

Appendix A.15
JCL/SAS Code: CMB3133

```
IN28.WKD833
IN29.WKD831
IN30.WKD833
IN31.WKD831
IN32.WKD833
IN33.WKD831
IN34.WKD833
IN35.WKD831
IN36.WKD833
IN37.WKD831
IN38.WKD833
IN39.WKD831
IN40.WKD833
IN41.WKD831
IN42.WKD833
IN43.WKD831
IN44.WKD833
IN45.WKD831
IN46.WKD833
;
BY SSN BIC DODEC;

/* DODEC IS THE DATE OF DECISION. THERE ARE MULTIPLE RECORDS FOR
BENEFICIARIES BECAUSE CDR'S (CONTINUING DISABILITY REVIEWS) ARE
OCCURRING EVERY 3 OR 7 YEARS, CREATING THE MULTIPLE RECORDS */
RUN;
PROC PRINT DATA=OUT1.COMB3133 (OBS=100);
FORMAT DODECM DOT DDPBM YYMMN6.; RUN;
```

Appendix A.16
JCL: F831ABAT

```
//$2358MIP JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358  
//*****  
//*  
//*          *-----*  
//*          | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(F831ABAT) |  
//*          *-----*  
//*  
//*          1. ALL STEPS ARE STEP RESTARTABLE  
//*          2. USES SAS PROGRAM CODE IS FROM SAME LIBRARY  
//*             IN A MEMBER NAMED PROC3133  
//*  
//* CONTACT NATALIE HAZELWOOD  
//* SSA PHONE -202 358-6228 MPR PHONE 202 264-3447  
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM  
//*  
//* PROCESS COMBINED 831-33 RECORDS -JCL CALLS PROGRAM FOR STEP1  
//*****  
//*  
//JS010    EXEC SAS9,  
//          WORK='120000,60000'  
//*  
//TEMP1    DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
//TEMP     DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
//IN       DD DSN=OPDR.TG.PRD.ETTW.$2358.CMB3133.Y2010.SSD,  
//          DISP=SHR  
//OUT2     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.SSICDR.SSD,  
//          DISP=(OLD,CATLG,DELETE),  
//          UNIT=TSILO  
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(PRO3133A),  
//          DISP=(SHR,PASS,KEEP)  
//*
```

Appendix A.17
SAS Code: PRO3133A

```
/* *****  
*****  
* FILENAME: OPDR.TG.PRD.ETTW.N8043.LIB07(PRO3133A)  
* PROGRAMMER:MIRIAM LOEWENBERG  
* MODIFIED BY: LESLEY & DAWN TO CREATE MIP, MINE VARIABLES  
* MARCH 2006  
* THIS SECTION OUTPUTS ONLY SSI FILE  
* PROC3133 HAD TO BE BROKEN DOWN INTO 2 PARTS TO GET IT TO RUN  
* PURPOSE:To create MONTHLY VARS FROM 831-833 FILES FOR TTW  
* T16 - CREATE SPANS DIAG AGE18_CDR -- CREATED FOR T2 IN MBR FILE  
* T16-T2 CREATE CDR DATES MIE ED  
* CREATED:02/26/05  
* LAST EDITED: 05/01/08  
* UPDATED FOR TRF10: NATALIE HAZELWOOD 10/25/2011  
*****  
/*  
* BELOW IS FOR TEST RUN ;  
*OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY OBS=5000;  
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY ;  
%LET BEGYR=1994;  
%LET ENDYR=2010; /* CHANGE AS NEEDED */  
%LET ENDMN=12; /* CHANGE AS NEEDED */  
  
/* STEP TO ASSIGN MACRO VARIABLES TO HANDLE TIME SERIES DATA */  
  
%MACRO START;  
%LET K=1;  
%DO I=&BEGYR %TO &ENDYR;  
%IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);  
%ELSE %IF %EVAL(&I-2000)<10 %THEN %LET YR=0%EVAL(&I-2000);  
%ELSE %LET YR=%EVAL(&I-2000);  
%IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);  
%ELSE %LET EMN=12;  
%DO J=1 %TO &EMN;  
%IF &J<10 %THEN %LET MN=0%EVAL(&J);  
%ELSE %LET MN=%EVAL(&J);  
%LET X&K=%EVAL(&YR)%EVAL(&MN);  
%LET K=%EVAL(&K+1);  
%END;  
%END;  
%LET TOT=%EVAL(&K-1);  
  
DATA CDR (KEEP=CDRTOT CDRT16 )  
TEMP.SSICDR /*SET ASIDE T16 RECORDS FOR SEPARATE PROCESSING */  
TEMP1.ALLCDR;  
SET IN.COMB3133;  
BY SSN BIC DODEC;  
  
/* THE COMBINATION OF SSN AND BIC IS THE LINK FOR T2 CASES.  
SSN FOR T2 IS THE CAN. ALL T2 RECORDS FOR A GIVEN CAN  
HAVE IDENTICAL SSN'S AND CAN BE DISTINGUISED ONLY BY THE BIC.  
THERE IS NO BOAN FIELD IN THE 831-33 FILES.  
  
THE LINK FOR T16 IS SSN - THE PAN - WHICH IS THE SAME AS
```

Appendix A.17

SAS Code: PRO3133A

SSN IN TRF. THE DATA LISTS ALL T16 BICS AS "A" EVEN THOUGH
BIC IS NOT RELEVANT TO T16 CASES. THE SSN'S ARE UNIQUE THOUGH
THERE MAY BE MULTIPLE RECORDS FOR A GIVEN BENEFICIARY */

```
/* COUNT UP TOTAL CDR'S-NUMBER T2 CDRS-NUMBER T16 CDRS.  
THESE FIELDS WILL INDEX THE ARRAYS FOR VARIABLES */
```

```
LENGTH CDRTOT CDRT16 CDRT2 3;  
RETAIN CDRTOT CDRT16 CDRT2;
```

```
/* RID R OR S FLAG SSI RECORDS - RID 2 OR 4 FLAG DI RECORDS */  
IF FIRST.BIC THEN DO;  
  CDRTOT = 1;  
  CDRT16 = 0;  
  CDRT2 = 0;  
  IF RID IN ('R' 'S') THEN CDRT16 = 1;  
  IF RID IN ('2' '4') THEN CDRT2 = 1;  
END;  
ELSE DO;  
  CDRTOT + 1;  
  IF RID IN ('R' 'S') THEN CDRT16 + 1;  
  IF RID IN ('2' '4') THEN CDRT2 + 1;  
END;  
OUTPUT TEMP1.ALLCDR;  
IF RID IN ('R' 'S') THEN OUTPUT TEMP.SSICDR;  
IF LAST.BIC THEN OUTPUT CDR;  
RUN;
```

```
/* PROCESS T16 DATA */  
PROC MEANS DATA=CDR NOPRINT;  
VAR CDRT16;  
OUTPUT OUT=LAST (DROP=_)  
      MAX=MAXT16;  
RUN;
```

```
PROC PRINT DATA=LAST;TITLE 'SSICDR'; RUN;
```

```
/* CREATE A MACRO VARIABLE TO INDEX THE FOLLOWING ARRAYS */  
/* IT FLAGS THE MAXIMUM NUMBER OF RECORDS FOR A BENEFICIARY */
```

```
DATA _NULL_;  
SET LAST;  
CALL SYMPUT('INX',MAXT16);  
STOP;  
RUN;
```

```
DATA OUT2.SSICDR  
(KEEP=SSN AGE18_CDR AGE18REDDT  
  %DO I = 1 %TO &INX; T16APPL&I  
                      T16START&I  
                      T16STOP&I  
                      T16BEGSPAN&I
```

Appendix A.17
SAS Code: PRO3133A

```
T16ENDSPAN&I
DEC&I T16RID&I %END;
%DO I = 1 %TO &TOT; T16DX1&&X&I T16DX2&&X&I %END;)
;
SET TEMP.SSICDR;
BY SSN BIC DODEC ; /* BIC IS ALWAYS "A" FOR T16 CLAIMS */

LENGTH
%DO I = 1 %TO &INX; T16APPL&I
T16START&I
T16STOP&I
T16BEGSPAN&I
T16ENDSPAN&I
%END; 4;

LENGTH
%DO I = 1 %TO &INX;
DEC&I T16RID&I
%END; $1;

LENGTH
%DO I = 1 %TO &TOT; T16DX1&&X&I T16DX2&&X&I
%END; $4;

/* SET UP ARRAYS FOR CREATING VARIABLES */
/* CREATE ARRAYS FOR MONTHLY VARIABLES */

ARRAY DX1{*} $4 %DO I=1 %TO &TOT; T16DX1&&X&I %END;;
ARRAY DX2{*} $4 %DO I=1 %TO &TOT; T16DX2&&X&I %END;;
ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;

RETAIN
%DO I=1 %TO &TOT; T16DX1&&X&I
T16DX2&&X&I %END;;

ARRAY APPL (*) %DO I = 1 %TO &INX; T16APPL&I %END;;
ARRAY STRT (*) %DO I = 1 %TO &INX; T16START&I %END;;
ARRAY STP (*) %DO I = 1 %TO &INX; T16STOP&I %END;;
ARRAY DEC (*) $ %DO I = 1 %TO &INX; DEC&I %END;;
ARRAY RD (*) $ %DO I = 1 %TO &INX; T16RID&I %END;;

RETAIN %DO I = 1 %TO &INX; T16APPL&I
T16START&I
T16STOP&I DEC&I T16RID&I %END; INDX;

LENGTH AGE18_CDR 3;
RETAIN AGE18_CDR AGE18REDDT;
RETAIN TERMIND;

IF FIRST.SSN THEN DO; /* BIC IS ALWAYS "A" FOR SSI RECORDS */

/* INITIALIZE DIAG VARS TO MISSING */
DO I = 1 TO &TOT;
DX1(I) = ' ';
DX2(I) = ' ';
```

Appendix A.17
SAS Code: PRO3133A

```
END;
TERMIND = 0;

DO I = 1 TO DIM(APPL);
  APPL(I) = .;
  STRT(I) = .;
  STP(I) = .;
  DEC(I) = ' ';
  RD(I) = ' ';
END;
AGE18_CDR = 0;
AGE18REDEDET = .;
INDX = 1;
DEC(INDX) = RDT;
RD(INDX) = RID;
IF AL IN ('B' 'C' 'D' 'E' 'F') THEN APPL(INDX) = DODECM;
IF DDPBM GT .Z THEN STRT(INDX) = DDPBM;
IF RDT = 'T' THEN STP(INDX) = DOT;
/* DETERMINE IF BENE HAS BEEN RE-DETERMINED AS ADULT */
IF DECADE GE 18 AND RDT = 'C' AND CDT = '04' AND WRM = '29'
  THEN DO;
  AGE18_CDR =1;
  AGE18REDEDET = DODEC;
  END;
END; /* FIRST SSN */

ELSE DO;
  INDX = INDX + 1;
  DEC(INDX) = RDT;
  RD(INDX) = RID;
  IF AL IN ('B' 'C' 'D' 'E' 'F') THEN APPL(INDX) = DODECM;
  IF DDPBM GT .Z THEN STRT(INDX) = DDPBM;
  IF RDT = 'T' THEN STP(INDX) = DOT;

  /* DETERMINE IF BENE HAS BEEN RE-DETERMINED AS ADULT */
  IF DECADE GE 18 AND RDT = 'C' AND CDT = '04' AND WRM = '29'
    THEN DO;
    AGE18_CDR =1;
    AGE18REDEDET = DODEC;
    END;
END; /* NOT FIRST SSN */

/* THE FOLLOWING ROUTINE CREATES MONTHLY REFERENCE DATES FROM
  JAN 1994 TO DEC 2007 */

J=1;
Y=&BEGYR;

DO I = 1 TO DIM(CYEAR); /* LOOP THROUGH ALL MONTH/YEAR REFERENCES
*/
  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
```

Appendix A.17
SAS Code: PRO3133A

```
IF INT(I/12)=I/12 THEN DO;
  J=1;
  Y=Y+1;
END;
ELSE J=J+1;

NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4) || SUBSTR(CYEAR(I),5,2),YVMN6.);

/* IF THERE IS A TERMINATION DATE - END MONTHLIES THEN */
IF DOT = NYEAR(I) THEN TERMIND = I;

IF (DDPBM > .Z AND (DDPBM <= NYEAR(i)) OR (DODECM > .Z
  AND DODECM <= NYEAR(I))) THEN DO;

  /* SELECT ACCEPTED CLAIMS */
  IF (RID = 'R' AND RDT = 'A') OR (RID = 'S' AND RDT = 'C') THEN DO;

    DX1(I) = RPDJ;
    DX2(I) = RSDJ;

  END; /* ACCEPTED CLAIMS */
  END; /* FALLS WITHIN MONTHLY RANGE */
END; /* LOOP THROUGH MONTHS */

IF LAST.SSN THEN DO;

  /* BLANK OUT THE VARIABLES AFTER TERMINATION DATE */
  IF DOT NE . THEN DO; /* ONLY IF THE LAST IS TERMINATED */
  DO I = 1 TO DIM(DX1);
    IF TERMIND = I THEN DO J = I TO DIM(DX1);
      DX1(J) = ' ';
      DX2(J) = ' ';
    END; /* TERMIND = I */
  END; /* DO I=1 TO DIM(DX1) */
END; /* DOT NE . */

/* DELETE STOP DATES IF A SUBSEQUENT APPEAL RESULTS IN CONTINUATION
*/
DO I = 2 TO &INX; /* ONLY IF MORE THAN ONE CLAIM */

  IF APPL(I) GT .Z AND DEC(I) = 'C' THEN DO; /* CONTINUATION */
  IF STP(I-1) NE . THEN STP(I-1) = .;
  END;

END; /* DO I = 2 TO &INX */

/* SET UP ARRAYS FOR SPANS OF ELIGIBILITY */

ARRAY BEGD (*) %DO I = 1 %TO &INX; T16BEGSPAN&I %END;;
ARRAY ENDD (*) %DO I = 1 %TO &INX; T16ENDSPAN&I %END;;
```

Appendix A.17
SAS Code: PRO3133A

```
FLGSTART = 0;
FLGEND = 0;
T16SPANS = 0;

DO I = 1 TO &INX;
  IF STRT(I) GT .Z AND ((RD(I) = 'R' AND DEC(I) = 'A')
                       OR (RD(I) = 'S' AND DEC(I) = 'C'))

  THEN DO;
    FLGSTART + 1;
    T16SPANS + 1;
    BEGD(FLGSTART) = STRT(I);
  END; /* START DATE AND ACCEPTED CLAIM */
  ELSE IF DEC(I) IN ('T' 'D') AND STP(I) GT .Z THEN DO;
    FLGEND + 1;
    ENDD(FLGEND) = STP(I);
  END;
END;

  /* SOME CASES HAVE NO START DATE GIVEN BY THE DISABILITY
ENTITLEMENT
  DATE. FOR THESE WE WILL USE THE DODEC TO CREATE 1 SPAN AND CHECK
  FOR A TERMINATION DATE */

  IF T16SPANS = 0 THEN DO;
    IF (RID = 'R' AND RDT = 'A')
      OR (RID = 'S' AND RDT = 'C') THEN DO; /* ACCEPTED CLAIMS */
      T16SPANS = 1;
      BEGD(1) = DODECM;
    END;
  END;

LABEL AGE18_CDR = "INDICATOR FOR ADULT REDETERMINATION"
AGE18REDET = "ADULT REDETERMINATION DATE"
%DO I=1 %TO &TOT;
  T16DX1&&x&i = "&&x&i T16 PRIMARY DIAGNOSIS"
  T16DX2&&x&i = "&&x&i T16 SECONDARY DIAGNOSIS"
%END;
%DO I=1 %TO &INX;
  T16APPL&I = "T16 APPEAL DATE &I"
  T16START&I = "T16 BENEFIT ENTITLEMENT DATE &I"
  T16STOP&I = "T16 BENEFIT CESSATION DATE &I"
  T16BEGSPAN&I = "T16 BEGINNING ENTITLEMENT DATE &I"
  T16ENDSPAN&I = "T16 CESSATION ENTITLEMENT DATE &I"
  DEC&I = "T16 RESULT OF DETERMINATION &I"
  T16RID&I = "T16 PROGRAM IDENTIFICATION &I"
%END;
;
OUTPUT;
END; /* LAST.SSN */
RUN;
PROC PRINT DATA=OUT2.SSICDR (OBS=50);
  TITLE 'SSI DATA';
  FORMAT %DO I = 1 %TO &INX;
```

Appendix A.17
SAS Code: PRO3133A

```
T16BEGSPAN&I
T16ENDSPAN&I %END; YYMMN6.;
RUN;
PROC FREQ DATA=OUT2.SSICDR; TABLES AGE18REDDT; WHERE AGE18_CDR = 1;
  FORMAT AGE18REDDT YYMMDD10.;
RUN;
PROC CONTENTS; RUN;

%MEND;
%START;
```

Appendix A.18
JCL: F831BBAT

```
// $2358MIP JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358  
// *****  
// *  
// *      *-----*  
// *      | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(F831BBAT) |  
// *      *-----*  
// *  
// *      1. ALL STEPS ARE STEP RESTARTABLE  
// *      2. USES SAS PROGRAM CODE IS FROM SAME LIBRARY  
// *          IN A MEMBER NAMED PROC133B  
// *  
// * CONTACT NATALIE HAZELWOOD  
// * SSA PHONE -202 358-6228 MPR PHONE 202 264-3447  
// * E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM  
// *  
// * PROCESS COMBINED 831-33 RECORDS - OUTPUT ALLCDR FILE  
// * JCL CALLS THE STEP 2 PROGRAM  
// *****  
// *  
// JS010    EXEC SAS9,  
//          WORK='120000,60000'  
// *  
// TEMP1    DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
// TEMP      DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
// IN        DD DSN=OPDR.TG.PRD.ETTW.$2358.CMB3133.Y2010.SSD,  
//          DISP=SHR  
// * CHANGE TO TAPE FOR PRODUCTION  
// * PRODUCTION JCL BELOW  
// OUT1      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALLCDR.SA.V1,  
//          DISP=(OLD,CATLG,DELETE),  
//          UNIT=TSILO  
// SYSIN     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(PRO3133B),  
//          DISP=(SHR,PASS,KEEP)  
// *
```

Appendix A.19
SAS Code: PRO3133B

```

/*****
* FILENAME: OPDR.TG.PRD.ETTW.N8043.LIB07(PRO3133B)
* PROGRAMMER:MIRIAM LOEWENBERG
* MODIFIED BY: LESLEY & DAWN TO CREATE MIP, MINE VARIABLES
* MARCH 2006
* THIS SECTION OUTPUTS THE ALL CDR - SSI-SSDI ALLCDR FILE
* PROC3133 HAD TO BE BROKEN INTO 2 PARTS TO GET IT TO RUN
* PURPOSE:To create MONTHLY VARS FROM 831-833 FILES FOR TTW
/811
* T16 - CREATE SPANS DIAG AGE18_CDR -- CREATED FOR T2 IN MBR FILE
* T16-T2 CREATE CDR DATES MIE ED DPM (NEW VARIABLE)
* CREATED:02/26/05
* LAST EDITED: 02/09/07

*****/
* BELOW IS FOR TEST RUN ;
*OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY OBS=5000;
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY ;
%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%LET ENDMN=12; /* CHANGE AS NEEDED */

/* STEP TO ASSIGN MACRO VARIABLES TO HANDLE TIME SERIES DATA */

%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
%IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
%ELSE %IF %EVAL(&I-2000)<10 %THEN %LET YR=0%EVAL(&I-2000);
%ELSE %LET YR=%EVAL(&I-2000);
%IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
%ELSE %LET EMN=12;
%DO J=1 %TO &EMN;
%IF &J<10 %THEN %LET MN=0%EVAL(&J);
%ELSE %LET MN=%EVAL(&J);
%LET X&K=%EVAL(&YR)%EVAL(&MN);
%LET K=%EVAL(&K+1);
%END;
%END;
%LET TOT=%EVAL(&K-1);

DATA CDR (KEEP=CDRTOT CDRT16 )
TEMP.SSICDR /*SET ASIDE T16 RECORDS FOR SEPARATE PROCESSING */
TEMP1.ALLCDR;
SET IN.COMB3133;
BY SSN BIC DODEC;

/* THE COMBINATION OF SSN AND BIC IS THE LINK FOR T2 CASES.
SSN FOR T2 IS THE CAN. ALL T2 RECORDS FOR A GIVEN CAN
HAVE IDENTICAL SSN'S AND CAN BE DISTINGUISHED ONLY BY THE BIC.
THERE IS NO BOAN FIELD IN THE 831-33 FILES.

THE LINK FOR T16 IS SSN - THE PAN - WHICH IS THE SAME AS
```

Appendix A.19
SAS Code: PRO3133B

SSN IN TRF. THE DATA LISTS ALL T16 BICS AS "A" EVEN THOUGH
BIC IS NOT RELEVANT TO T16 CASES. THE SSN'S ARE UNIQUE THOUGH
THERE MAY BE MULTIPLE RECORDS FOR A GIVEN BENEFICIARY */

/* COUNT UP TOTAL CDR'S-NUMBER T2 CDRS-NUMBER T16 CDRS.
THESE FIELDS WILL INDEX THE ARRAYS FOR VARIABLES */

LENGTH CDRTOT CDRT16 CDRT2 3;
RETAIN CDRTOT CDRT16 CDRT2;

IF FIRST.BIC THEN DO;
CDRTOT = 1;
CDRT16 = 0;
CDRT2 = 0;
IF RID IN ('R' 'S') THEN CDRT16 = 1;
IF RID IN ('2' '4') THEN CDRT2 = 1;
END;
ELSE DO;
CDRTOT + 1;
IF RID IN ('R' 'S') THEN CDRT16 + 1;
IF RID IN ('2' '4') THEN CDRT2 + 1;
END;
OUTPUT TEMP1.ALLCDR;
IF RID IN ('R' 'S') THEN OUTPUT TEMP.SSICDR;
IF LAST.BIC THEN OUTPUT CDR;

RUN;
PROC CONTENTS DATA=TEMP1.ALLCDR;
RUN;
/* GET THE MAXIMUM NUMBER OF CDR'S FOR ARRAYS */

PROC MEANS DATA=CDR NOPRINT;
VAR CDRTOT;
OUTPUT OUT=LAST (DROP=_)
MAX=MAXCDR;
RUN;
PROC CONTENTS; TITLE 'LAST'; RUN;
PROC PRINT DATA=LAST;TITLE 'ALLCDR'; RUN;
DATA _NULL_;
SET LAST;
CALL SYMPUT('IND',MAXCDR);
STOP;
RUN;

DATA OUT1.ALLCDR
(KEEP= SSN BIC PROGRAM
%DO I = 1 %TO &IND; CDRDTE&I JUDLVL&I RDT&I RID&I RB&I RFC&I
DPM&I
%END;
%DO I = 1 %TO &TOT;
MIEX&&X&I
ALX&&X&I
EDX&&X&I

Appendix A.19
SAS Code: PRO3133B

```
                                %END; )
                                ;

LENGTH
  %DO I = 1 %TO &IND;
    JUDLVL&I
    RDT&I
    RID&I
    DPM&I
  %END;
  $1;
SET TEMP1.ALLCDR;
BY SSN BIC DODEC ;

/* CREATE A FLAG TO INDICATE DI SSI AND CONCURRENTS -
WE NEED THIS TO LINK BACK TO TRF. FIRST LINK UP THE DI
USING CAN AND BIC (DI ONLIES). FOR SSI
USE THIS SSN SINCE THEY WILL BE SSI ONLIES.
FOR CONCURRENTS FLAG BOTH PROGRAMS SINCE WE WILL BE
COMBINING ANY SSI-DI DATA IN THE GET2SSN PROGRAM */

RETAIN PROGRAM1 PROGRAM2;
LENGTH PROGRAM1 PROGRAM2 3;

IF FIRST.BIC THEN DO;

  PROGRAM1 = 0; PROGRAM2 = 0;
  IF RID IN('2' '4') THEN PROGRAM1 = 1;
  ELSE IF RID IN('R' 'S') THEN PROGRAM2 = 1;
END;
ELSE DO;
  IF RID IN('2' '4') THEN PROGRAM1 = 1;
  ELSE IF RID IN('R' 'S') THEN PROGRAM2 = 1;
END;

/* SET UP ARRAYS FOR CREATING VARIABLES */

ARRAY CDRDT (*) %DO I = 1 %TO &IND; CDRDTE&I %END;;
ARRAY CDRDTE&I %DO I = 1 %TO &IND; JUDLVL&I %END;;
ARRAY DECIS ($) %DO I = 1 %TO &IND; RDT&I %END;;
ARRAY TERM (*) %DO I = 1 %TO &IND; TRMDTE&I %END;;
ARRAY PROG ($) %DO I = 1 %TO &IND; RID&I %END;;
ARRAY DIS (*) %DO I = 1 %TO &IND; DDPBM&I %END;;
ARRAY REG ($) %DO I = 1 %TO &IND; RB&I %END;;
ARRAY CONTIN ($) %DO I = 1 %TO &IND; RFC&I %END;;
ARRAY PERM ($) %DO I = 1 %TO &IND; DPM&I %END;;

RETAIN %DO I = 1 %TO &IND; CDRDTE&I %END;;
RETAIN %DO I = 1 %TO &IND; JUDLVL&I %END;;
RETAIN %DO I = 1 %TO &IND; RDT&I %END;;
RETAIN %DO I = 1 %TO &IND; RID&I %END;;
RETAIN %DO I = 1 %TO &IND; TRMDTE&I %END;;
RETAIN %DO I = 1 %TO &IND; DDPBM&I %END;;
```

Appendix A.19
SAS Code: PRO3133B

```
RETAIN %DO I = 1 %TO &IND; RB&I %END;;  
RETAIN %DO I = 1 %TO &IND; RFC&I %END;;  
RETAIN %DO I = 1 %TO &IND; DPM&I %END;;  
RETAIN INDX;
```

```
IF FIRST.BIC THEN DO;  
  /* INITIALIZE VARIABLES */  
  DO I = 1 TO DIM(CDRDT);  
    CDRDT(I) = .;  
    CDRTYP(I) = ' '  
    DECIS(I) = ' '  
    PROG(I) = ' '  
    TERM(I) = .;  
    DIS(I) = .;  
    REG(I) = ' '  
    CONTIN(I) = ' '  
    PERM(I) = ' '  
  END;
```

```
  INDX = 1;  
  CDRDT(INDX) = DODEC;  
  CDRTYP(INDX) = AL;  
  DECIS(INDX) = RDT;  
  PROG(INDX) = RID;  
  TERM(INDX) = DOT;  
  DIS(INDX) = DDPBM;  
  REG(INDX) = RB;  
  CONTIN(INDX) = RFC;  
  PERM(INDX) = DPM;  
END;
```

```
ELSE DO;  
  INDX = INDX + 1;  
  CDRDT(INDX) = DODEC;  
  CDRTYP(INDX) = AL;  
  DECIS(INDX) = RDT;  
  TERM(INDX) = DOT;  
  PROG(INDX) = RID;  
  DIS(INDX) = DDPBM;  
  REG(INDX) = RB;  
  CONTIN(INDX) = RFC;  
  PERM(INDX) = DPM;  
END;
```

```
LENGTH  
%DO I=1 %TO &TOT;  
  MIEX&&X&I  
  ALx&&x&i  
%END;  
$1;
```

```
LENGTH  
%DO I=1 %TO &TOT;  
  EDX&&X&I  
%END;  
$2;
```

Appendix A.19
SAS Code: PRO3133B

```
LENGTH
  %DO I = 1 %TO &IND;
    CDRDTE&I
  %END;
  4;

/* CREATE ARRAYS FOR MONTHLY VARIABLES FROM 1994 THRU 2007 */

ARRAY EDX{*} $ %DO I=1 %TO &TOT; EDX&&X&I %END;;
ARRAY ALX{*} $ %DO I=1 %TO &TOT; ALX&&X&I %END;;
ARRAY MIE{*} $ %DO I=1 %TO &TOT; MIE&&X&I %END;;
ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;

RETAIN
  %DO I=1 %TO &TOT; EDX&&X&I %END;
  %DO I=1 %TO &TOT; MIE&&X&I %END;
  %DO I=1 %TO &TOT; ALX&&X&I %END; ;

RETAIN TERMIND1 TERMIND2;

IF FIRST.BIC THEN DO;
  DO I = 1 TO &TOT;
    EDX(I) = ' ';
    MIE(I) = '0';
    ALX(I) = ' ';
  END;
  TERMIND1 = 0;
  TERMIND2 = 0;
END; /* FIRST BIC */

/* THE FOLLOWING ROUTINE CREATES MONTHLY REFERENCE DATES FROM
JAN 1994 TO DEC 2007 - EACH RECORD WILL BE EVALUATED.
FOR BENEFICIARIES WITH MULTIPLE RECORDS THE FIELDS ARE RETAINED
FROM RECORD TO RECORD */

J=1;
Y=&BEGYR;

DO I = 1 TO DIM(CYEAR); /* LOOP THROUGH ALL MONTH/YEAR REFERENCES */

  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
  IF INT(I/12)=I/12 THEN DO;
    J=1;
    Y=Y+1;
  END;
  ELSE J=J+1;

  NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4) || SUBSTR(CYEAR(I),5,2),YYMMN6.);

/* IF THE DISAB ENTIT MONTH EXISTS USE IT. ELSE USE DODEC(ADJUD
DATE)
  THE DISAB ENTIT MONTH USUALLY PREDATES THE DODEC,
```

Appendix A.19
SAS Code: PRO3133B

```
BUT IS OFTEN MISSING.
  COMPARE TO EACH MONTHLY REFERENCE VARIABLE TO SET MONTHLY VARS
*/

/* IF THERE IS A TERMINATION DATE - END MONTHLIES THEN */
/* LOOK AT SEPARATE TITLE TERMINATIONS */
/* FOR MULTIPLE TERMINATIONS ON THE SAME PROGRAM THE LAST
WILL BE KEPT.
  SET TERMINATION INDEXES USING THE ARRAY REFERENCE I WHEN
  A TERMINATION DATE (DOT) IS IN THAT ARRAY MONTH */

IF DOT = NYEAR(I) AND RID IN ('2' '4') THEN TERMIND1 = I;
ELSE IF DOT = NYEAR(I) AND RID IN ('R' 'S') THEN TERMIND2 = I;

IF (DDPBM > .Z AND (DDPBM <= NYEAR(i)) OR (DODECM > .Z
  AND DODECM <= NYEAR(I))) THEN DO;

  EDX(I) = ED;
  /* CHANGE FOR 2006 - ALX EVALUATED FOR ACCEPTED AND DENIED CLAIMS
*/

  ALX(I) = AL;

IF (RID IN ('2' 'R') AND RDT = 'A') OR
  (RID IN ('4' 'S') AND RDT = 'C') THEN DO; /* ACCEPTED CLAIMS */

  /* NEW DEFINITION OF MIE */
  IF DR IN ('1', '3', '9') THEN MIE(I)='P';
  ELSE IF DR IN ('5', '7') THEN MIE(I)='N';
  ELSE IF DR IN ('E', 'F', 'G', 'I', 'J', 'K', 'L', 'M', 'N', 'V', 'W', 'X',
    'Y', 'Z') THEN DO;
    IF DPM='P' THEN MIE(I)='N';
    ELSE MIE(I)='P';
  END;
  ELSE IF 'A' <= DR <= 'Z' THEN MIE(I)='E';

  END; /* ACCEPTED CLAIMS */
END; /* DATE FALLS WITHIN MONTH */

END; /* LOOP THROUGH ALL MONTH-YEAR REFERENCES */

IF LAST.BIC THEN DO;

  /* TAKE THE LATEST TERMINATION TO BLANK OUT VARIABLES */
  /* IF FOR A TITLE THERE IS NO TERMINATION THEN DO NOT BLANK */
  /* IF BOTH TITLES THEN TAKE THE LATEST */

  IF CDRT2 GT 0 AND CDRT16 GT 0 THEN DO; /* BOTH PROGRAMS */

  IF TERMIND1 GT 0 AND TERMIND2 GT 0 THEN DO;
    /* TERMINATION ON BOTH PROGRAMS */
    IF TERMIND1 GT TERMIND2 THEN TERMIND = TERMIND1;
    ELSE TERMIND = TERMIND2;
  END;
```

Appendix A.19
SAS Code: PRO3133B

```
/* DO NOTHING MORE IF BOTH PROGRAMS REPRESENTED. IF ONE PROGRAM
HAS NO TERMINATION THEN MONTHLY VARIABLES ARE CONTINUED ON THE
CLAIM NOT YET TERMINATED. THESE VARIABLES ARE NOT PROGRAM
SPECIFIC */
END;

ELSE DO; /* ONLY ONE PROGRAM REPRESENTED */
/* EITHER TERMINATION CLOSES THE BENEFICIARIES ACCOUNT */
IF TERMIND1 GT 0 THEN TERMIND = TERMIND1;
ELSE IF TERMIND2 GT 0 THEN TERMIND = TERMIND2;
END;

/* BLANK OUT THE VARIABLES AFTER TERMINATION DATE */
/* ONLY IF THE LAST CLAIM IS TERMINATED */

IF DOT NE . THEN DO;

DO I = 1 TO DIM(MIE);
IF TERMIND = I THEN DO J = I TO DIM(MIE);
ALX(J) = ' ';
MIE(J) = '0';
END;
END;

END; /* DOT NE . */

LENGTH PROGRAM 3;

/* LAST BIC - IF BIC = A AND HE IS NOT A CONCURRENT
HE WILL BE ASSIGNED TO DI AS A CAN.
HOWEVER FOR DI PRIMARIES (BIC = A) WHO ARE CONCURRENTS
RECORDS MAY APPEAR IN THE SSI SEGMENT. THE INFORMATION ON THESE
RECORDS MUST BE COMBINED WITH THE DI INFORMATION IN GET2SSN
BEFORE THEY ARE DE-DUPED.
A WIDOW OR A DAC WHO HAS A CAN WITH A BIC OF C OR W WILL
BE OUTPUT IN PROGRAM 1 AS A CAN. BUT THERE COULD BE AN SSI
RECORD
FOR THE PRIMARY UNDER A PAN AND A BIC = A AND THIS CASE
WOULD BE OUTPUT AS PROGRAM 2 SSI. THESE RECORDS WILL BE
SCREENED
OUT IN THE GET2SSN PROGRAM WHERE ONLY SSI RECORDS IN THE FINDER
FOR SSI WILL BE KEPT */

IF PROGRAM1 = 1 AND PROGRAM2 = 0 THEN PROGRAM = 1; /*T2 ONLY CASES */
ELSE IF PROGRAM1 = 1 AND PROGRAM2 = 1 THEN PROGRAM = 2; /* CONCURRENTS
*/
ELSE IF PROGRAM2 = 1 THEN PROGRAM = 3; /* REMAINDER T16 ONLIES */

LABEL
%DO I=1 %TO &TOT;
MIEX&&x&i = "&&x&i MEDICAL IMPROVEMENT EXPECTED"
ALX&&x&i = "&&x&i LEVEL OF ADJUDICATION"
EDX&&x&i = "&&x&i YEARS OF EDUCATIOM"
```

Appendix A.19
SAS Code: PRO3133B

```
%END;
%DO I=1 %TO &IND;
  CDRDTE&I = "CDR DATE &I"
  JUDLVL&I = "LEVEL OF ADJUDICATION &I"
  RDT&I = "RESULT OF DETERMINATION &I"
  RID&I = "PROGRAM IDENTIFICATION &I"
  RB&I = "REGULATION BASIS CODE &I"
  RFC&I = "REASON FOR CONTINUATION-CESSATION CODE &I"
  DPM&I = "PERMANENT DISABILITY CODE &I"
%END;
;
OUTPUT OUT1.ALLCDR;

END; /* LAST BIC */
RUN;

PROC FREQ DATA=OUT1.ALLCDR;
TABLES PROGRAM * MIEX0001
        PROGRAM * MIEX0101
        PROGRAM * MIEX0201
        PROGRAM * MIEX0301
        PROGRAM * MIEX0401
        PROGRAM * MIEX0501
        PROGRAM * MIEX0601
;
RUN;

PROC PRINT DATA=OUT1.ALLCDR (OBS=100);
  TITLE 'CDR DATA ALL';RUN;
PROC CONTENTS; RUN;

%MEND;
%START;
```

Appendix A.20

JCL/SAS Code: GET2SSN1

```

//$2358GET JOB (11710000,T715,,SAS,,ITC9FL),2358MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.$2358.TRF10.P831.PRDLIB(GET2SSN1) |
//*          *-----*
//* RETRIEVE THE SSN FROM CAN/BIC COMBO FOR T2 RECORDS
//* ATTACH SSN TO ALLCDR RECORDS PRELIMINARY TO MERGING WITH TRF
//* THIS PROGRAM IS STEP1 AND OUTPUTS THE DATA FILES TO BE
//* INPUT IN STEP 2 (GET2SSN2)
//* DATE: 4/14/10
//* CONTACT NATALIE HAZELWOOD
//* SSA PHONE 202 358-6228 MPR PHONE 202 264-3447
//* E-MAIL NHAZELWOOD@MATHEMATICA-MPR.COM
//*****
//*
// SET REG='64M'
//JS010 EXEC SAS9,
//      WORK='200000,100000',          * SPACE IN BLKS *
//      REGION=&REG,
//      PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALLCDR.SA.V1,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.$2358.ALLCDR1.SSI10.SA,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//OUT2 DD DSN=OPDR.TG.PRD.ETTW.$2358.ALLCDR1.SSD10.SA,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//SYSIN DD *
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;
/* CHECK RECORDS IN THE LINKING FILE */
PROC PRINT DATA=IN2.LINKDI (OBS=25); TITLE 'DI LINK'; RUN;

/* SPLIT ALLCDR FILE BY PROGRAM */
DATA TEMP.SSI TEMP1.DI (RENAME=(SSN=CAN));
SET IN1.ALLCDR;
/* OUTPUT THE CONCURRENTS ON BOTH SSI AND DI DATA SEGMENTS */
IF PROGRAM IN (1 2) THEN OUTPUT TEMP1.DI;
IF PROGRAM IN (2 3) THEN OUTPUT TEMP.SSI;
RUN;

/* SORT SSI RECORDS BY SSN (PAN) IN PREPARATION FOR FINAL MERGE */
PROC SORT DATA=TEMP.SSI OUT=OUT1.SSI; BY SSN; RUN;

```

Appendix A.20
JCL/SAS Code: GET2SSN1

```
PROC SORT DATA=TEMP1.DI OUT=OUT2.DI; BY CAN BIC; RUN;
```

Appendix A.21
JCL/SAS Code: GET2SSN2

```
// $8043GET JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *****
// *
// *      *-----*
// *      | OPDR.TG.PRD.ETTW.N8043.LIB11(GET2SSN2) |
// *      *-----*
// * RETRIEVE THE SSN FROM CAN/BIC COMBO FOR T2 RECORDS
// * ATTACH SSN TO ALLCDR RECORDS PRELIMINARY TO MERGING WITH TRF
// * STEP 2 - INPUT FILES OUTPUT IN STEP 1 (GET2SSN1)
// * DATE: 2/22/07
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// SET REG='64M'
// JS010 EXEC SAS9,
//       WORK='200000,100000',          * SPACE IN BLKS *
//       REGION=&REG,
//       PARM='MEMSIZE=&REG'
// *
// INS DD DSN=OPDR.TG.PRD.ETTW.$2358.ALLCDR1.SSI10.SA,DISP=SHR
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
// IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,DISP=SHR
// IND DD DSN=OPDR.TG.PRD.ETTW.$2358.ALLCDR1.SSD10.SA,DISP=SHR
// TEMP DD DSN=&&TEMP,
//       DISP=(NEW,DELETE,DELETE),
//       SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// TEMPS DD DSN=&&TEMP,
//       DISP=(NEW,DELETE,DELETE),
//       SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// TEMP1 DD DSN=&&TEMP,
//       DISP=(NEW,DELETE,DELETE),
//       SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// TEMP2 DD DSN=&&TEMP,
//       DISP=(NEW,DELETE,DELETE),
//       UNIT=TSILO
// TEMP3 DD DSN=&&TEMP,
//       DISP=(NEW,DELETE,DELETE),
//       UNIT=TSILO
// OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,
//       DISP=(NEW,CATLG,DELETE),
//       UNIT=TSILO
// SYSIN DD *
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;
/* SUBSET SSI DATA TO SSN'S IN SSI FINDER. */

DATA FINDER;
  INFILE IN3 TRUNCOVER;
  INPUT @001 SSN $9.;
RUN;
PROC SORT; BY SSN; RUN;
```

Appendix A.21
JCL/SAS Code: GET2SSN2

```
DATA TEMP.SSI;
  MERGE INS.SSI(IN=A) FINDER (IN=B);
  BY SSN;
  IF A AND B;
  RUN;
PROC SORT DATA=TEMP.SSI OUT=TEMPS.SSI NODUPKEY; BY SSN; RUN;

PROC SORT DATA=IN2.LINKDI OUT=LINKDI;
  BY CAN BIC;
  RUN;

  /* SUBSET DI DATA TO THE CAN/BIC COMBINATIONS IN DI LINKING FILE */

DATA TEMP1.DI ;
  MERGE IND.DI (IN=D) LINKDI (IN=B);
  BY CAN BIC;
  IF D AND B THEN OUTPUT TEMP1.DI;
  RUN;

  /* NEW DATA STEP TO ELIMINATE THE DUPLICATE SSN'S PULLED BY CAN DATA
  IN THE 831/33 FILES - THE CAN AND BIC ARE THE IDENTIFIERS IN
831.
  OUR LINKING FILE HAS EACH CAN SSN AND BIC COMBINATIONS WHICH MAY
  REPEAT THE SSN FOR DUAL ELIGIBLES (EG. BENE WITH BIC OF A ON HER
  PRIMARY ACCOUNT WHO HAS A BIC OF W ON HER HUSBAND'S ACCOUNT WHERE
  SHE GETS WIDOW'S BENEFITS. WE WILL TAKE THE FIRST RECORD WHICH
WILL
  BE THE PRIMARY IF THERE IS A BIC = A RECORD. IF NOT THEN
  WE WILL TAKE THE FIRST AUXILIARY RECORD. THIS IS THE BEST WE CAN
DO
  FOR DUALS */

PROC SORT DATA=TEMP1.DI OUT=TEMP2.DI ;
  BY SSN BIC;
  RUN;

PROC DATASETS LIBRARY=TEMP1;
  DELETE DI;
  RUN;

DATA TEMPA.DI;
  SET TEMP2.DI;
  BY SSN BIC;
  /* IF THE CAN = A KEEP THE RECORD - OTHERWISE KEEP THE FIRST
```

Appendix A.21
JCL/SAS Code: GET2SSN2

```
AUXILIARY */
IF FIRST.SSN;
RUN;

/* COMBINE SSI AND DI SEGMENTS */

DATA OUT1.ALLCDR;
MERGE TEMPS.SSI TEMPA.DI;
BY SSN;
/* THERE ARE RECORDS FOR SOME DAC'S AND WIDOWS IN THE SSI SEGMENT -
  DUPLICATES OF RECORDS ON THE DI SIDE. IT SEEMS THAT AT SOME
  POINT A RECORD HAS BEEN CREATED IN SSI FOR SOME OF THESE BENES.
  THIS SHOWS UP AS A DUPLICATE BECAUSE THE BIC IS W OR C IN THE
  DI PROGRAM ON 831 AND A IN THE SSI PROGRAM. THESE DUP RECORDS
  HAVE BEEN SCRUTINIZED. IN MANY CASE THEY HAVE DUPLICATE VALUES
  FOR THE VARIABLES. WHEN THESE ARE NOT DUPLICATES THE DI RECORD
  USUALLY HAS MORE COMPLETE DATA. THEREFORE WE HAVE CHANGED THE
  DATA STEP FROM "SET" TO "MERGE" PLACING THE SSI RECORD FIRST
  TO BE OVERWRITTEN BY THE DI RECORD (SECOND) WHEN THE SSN
  APPEARS IN BOTH FILES. */
RUN;
PROC CONTENTS DATA=OUT1.ALLCDR; RUN;
PROC PRINT DATA=OUT1.ALLCDR (OBS=25); RUN;
/* CHECK FOR DUPS ON OUTPUT DATASET */
PROC SORT DATA=OUT1.ALLCDR(KEEP=SSN) OUT=ALLCDR NODUPKEY;
BY SSN;
RUN;
```

Appendix A.22
JCL: LOAD1JCL

```
// $8043LD1 JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *          RESTART=JS050.SAS9
// *****
// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.N8043.LIB10(LOAD1JCL) |
// *          *-----*
// * TRF09 RECORDS ARE IN 4 SECTIONS - FIRST LOAD JCL
// * PRELIMINARY TO ADDING UP BENEFITS
// * READ IN THE BENEFIT FIELDS FOR ALL DI BENEFICIARIES
// * JCL CALLS PROGRAM MBRLOAD1
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// JS010 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR1.R111011,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR1,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1),
//          DISP=(SHR,PASS,KEEP)
// *
// JS020 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR2.R111011,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR2,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1),
//          DISP=(SHR,PASS,KEEP)
// *
// JS030 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR3.R111011,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR3,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
```

Appendix A.22

JCL: LOAD1JCL

```
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1),
//      DISP=(SHR,PASS,KEEP)
//*
//JS040 EXEC SAS9,
//      WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR4.R111011,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR4,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1),
//      DISP=(SHR,PASS,KEEP)
//*
//JS050 EXEC SAS9,
//      WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR5.R111011,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR5,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1),
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.23
SAS Code: MRBLOAD1

```
/******  
  
      *-----*  
      | OPDR.TG.PRD.ETTW.N8043.LIB11(MBRLOAD1) |  
      *-----*  
  
      SASLOAD ALL RECORDS RETURNED FROM MBR FINDER  
      PRELIMINARY TO ADDING UP BENEFITS (MBRADDUP)  
      CONTACT MIRIAM LOEWENBERG  
      SSA PHONE 202 358-6214 MPR PHONE 202 484-4829  
      E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM  
*****/  
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY;  
%LET BEGYR=1994;  
%LET ENDYR=2010; /* CHANGE END YEAR AS NEEDED */  
%LET ENDMN=12; /* CHANGE END MONTH AS NEEDED */  
  
%macro start;  
%let k=1;  
  
%do i=&begyr %to &endyr;  
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);  
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);  
  %ELSE %LET YR=%EVAL(&I-2000);  
  %IF &I=&BEGYR %THEN %LET BMN=1; /*START AT JANUARY */  
  %ELSE %LET BMN = 1;  
  %if &i=&endyr %then %let emn=%eval(&endmn);  
  %else %let emn=12;  
  %do j=&BMN %to &emn;  
    %if &j<10 %then %let mn=0%eval(&j);  
    %else %let mn=%eval(&j);  
    %let x&k=%eval(&yr)%eval(&mn);  
    %let k=%eval(&k+1);  
  %end;  
%end;  
%let tot=%eval(&k-1);  
  
/* EXTRACT THE PAYMENT FIELDS FOR USE IN COMPUTING DEPENDENT AMTS */  
/* THE DATA WILL CONTAIN ALL AUXILIARIES */  
  
data TEMP.mbr;  
  
infile in1 trunccover;  
input  
  @1 can $9.  
  @61 BIC $2.  
  @305 boan $9.  
  @;  
  
POS = 6784;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS MBC&&X&I 5.1 /* Monthly Benefit Credited */
```

Appendix A.23
SAS Code: MRBLOAD1

```
@;  
  POS=POS+5;  
%END;  
POS = 9640;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS BPD&&X&I $1./* BENEFIT PAYMENT DESIGNATION */  
  @;  
  POS=POS+1;  
%END;  
;  
  
  IF SUBSTR(BIC,1,1) IN ('A') THEN SSN = CAN;  
  ELSE SSN = BOAN;  
  DROP POS;  
RUN;  
  /* CHECK THAT NO DUPLICATE RECORDS ARE IN THE FILE */  
PROC SORT DATA=TEMP.MBR OUT=OUT.MBR NODUPKEY; BY CAN BIC; RUN;  
  
PROC PRINT DATA=OUT.MBR (OBS=25);  
  TITLE 'MRBLOAD1';  
RUN;  
PROC FREQ DATA=OUT.MBR ; TABLES BIC; RUN;  
RUN;  
  
%MEND;  
  
%START;
```

Appendix A.24
JCL/SAS Code: LOAD1COM

```

//$8043LDC JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(LOAD1COM) |
//*          *-----*
//*
//*          1.  COMBINE EACH SEGMENT OF MBR LOAD1 FILE (DEPENDENT AMTS)
//*          2.  SORTED BY CAN AND BIC AND OUTPUT COMBINED FILE
//*
//* OUTPUT FILE WILL BE PROCESSED TO ADD UP DEPENDENT BENEFIT AMOUNTS
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR1,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR2,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR3,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR4,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBRLOAD1.MBR5,DISP=SHR
//*
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,500),RLSE),VOL=(,,10)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MBRLOAD1.COMB10.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,500),RLSE),VOL=(,,10)
//*
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY;
DATA TEMP.MBR;
  SET IN1.MBR
      IN2.MBR
      IN3.MBR
      IN4.MBR
      IN5.MBR;
  BY CAN BIC;

RUN;
PROC SORT DATA=TEMP.MBR OUT=OUT1.MBR NODUPKEY; BY CAN BIC; RUN;
PROC PRINT DATA=OUT1.MBR (OBS=50); RUN;
PROC FREQ DATA=OUT1.MBR; TABLES BIC; RUN;

```

Appendix A.25
JCL: LOPHUJCL

```
// $8043LD2 JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *          RESTART=JS050.SAS9
// *****
// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.N8043.LIB11(LOPHUJCL) |
// *          *-----*
// * PHUS RECORDS ARE IN 4 SECTIONS
// * PRELIMINARY TO ADDING UP BENEFITS
// * READ IN THE DIRECT PAY AND MEDICARE PREMIUMS
// * JCL CALLS PROGRAM PHUSLOAD
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// JS010 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUS1.R110920,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE1,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD),
//          DISP=(SHR,PASS,KEEP)
// *
// JS020 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUS2.R110920,DISP=SHR
// TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE2,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD),
//          DISP=(SHR,PASS,KEEP)
// *
// JS030 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUS3.R110920,DISP=SHR
```

Appendix A.25
JCL: LOPHUJCL

```
//TEMP DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//TEMP1 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE3 ,
//      DISP=(NEW,CATLG,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD) ,
//      DISP=(SHR,PASS,KEEP)
//*
//JS040 EXEC SAS9 ,
//      WORK='200000,100000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUS4.R110920,DISP=SHR
//TEMP DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//TEMP1 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE4 ,
//      DISP=(NEW,CATLG,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD) ,
//      DISP=(SHR,PASS,KEEP)
//*
//JS050 EXEC SAS9 ,
//      WORK='200000,100000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUS5.R110920,DISP=SHR
//TEMP DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//TEMP1 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE5 ,
//      DISP=(NEW,CATLG,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD) ,
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.26
SAS Code: PHUSLOAD

```

/*****
*-----*
|  OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSLOAD)  |
*-----*

SASLOAD PHUS RECORDS RETURNED FROM MBR FINDER
PRELIMINARY TO ADDING UP BENEFITS (MBRADDUP)
CONTACT MIRIAM LOEWENBERG
SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
*****/
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY;
%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE END YEAR AS NEEDED */
%LET ENDMN=12; /* CHANGE END MONTH AS NEEDED */

%macro start;
%let k=1;

%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&BEGYR %THEN %LET BMN=1; /*START AT JANUARY */
  %ELSE %LET BMN = 1;
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=&BMN %to &emn;
    %if &j<10 %then %let mn=0%EVAL(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

/* THE DATA WILL CONTAIN ALL AUXILIARIES */

data TEMP.PHUS;

infile in1 trunccover;
input
  @1 CAN $9.
  @10 BIC $2.
  @;
POS = 39;
%DO I=1 %TO 324;
  INPUT
  @POS DIRPAY&I ZD7.1 /* Direct Pay */
  @;
  POS=POS+7;
%END;

```

Appendix A.26
SAS Code: PHUSLOAD

```
POS = 2307;
%DO I=1 %TO 324;
  INPUT
  @POS MEDPREM&I ZD5.1/* Medicare Premium */
  @;
  POS=POS+5;
%END;
;

DROP POS;
RUN;

DATA TEMP1.PHUS;
SET TEMP.PHUS
  (KEEP=CAN BIC DIRPAY121-DIRPAY324 MEDPREM121-MEDPREM324);

  ARRAY PAY (*) %DO I=1 %TO &TOT; PAYD&&X&I %END;;
  ARRAY DPO (*) %DO I=1 %TO &TOT; DIRPAY&&X&I %END;;
  ARRAY MEDO (*) %DO I=1 %TO &TOT; MEDPREM&&X&I %END;;
  ARRAY DP (*) DIRPAY121-DIRPAY324;
  ARRAY MED(*) MEDPREM121-MEDPREM324;

  /* SUM DIRECT PAY AND MEDICARE PREMIUM TO GET THE TOTAL AMT PAID
*/

  DO I = 1 TO &TOT;
    DPO(I) = DP(I);
    MEDO(I) = MED(I);
    PAY(I) = SUM(DP(I),MED(I));
  END;

DROP DIRPAY121-DIRPAY312 MEDPREM121-MEDPREM312;

RUN;

  /* CHECK THAT NO DUPLICATE RECORDS ARE IN THE FILE */
PROC SORT DATA=TEMP1.PHUS OUT=OUT.PHUS NODUPKEY; BY CAN BIC;
RUN;

PROC PRINT DATA=OUT.PHUS (OBS=25);
  TITLE 'PHUSLOAD';
RUN;
PROC FREQ DATA=OUT.PHUS ; TABLES BIC; RUN;
RUN;

%MEND;

%START;
```

Appendix A.27
JCL/SAS Code: PHUSCOMB

```
//$8043LDC JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(PHUSCOMB) |
//*          *-----*
//*
//*          1. COMBINE EACH SEGMENT OF PHUS FILE
//*          2. SORTED BY CAN AND BIC AND OUTPUT COMBINED FILE
//*
//* OUTPUT FILE WILL BE PROCESSED TO ADD UP DEPENDENT BENEFIT AMOUNTS
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE1,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE2,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE3,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE4,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.PHUSLOAD.FILE5,DISP=SHR
//*
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.PHUS.COMB10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//*
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY;
DATA OUT1.PHUS;
  SET IN1.PHUS
      IN2.PHUS
      IN3.PHUS
      IN4.PHUS
      IN5.PHUS ;
  BY CAN BIC;
  IF LAST.BIC;
  /* DROP VARIABLES FORGOTTEN IN THE READPHUS PROGRAM */
  DROP DIRPAY313-DIRPAY324 MEDPREM313-MEDPREM324;
RUN;
PROC PRINT DATA=OUT1.PHUS (OBS=50); RUN;
PROC FREQ DATA=OUT1.PHUS; TABLES BIC; RUN;
```

Appendix A.28
JCL/SAS Code: MRGPHUS1

```

//$8043LDC JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB10(MRGPHUS) |
//*          *-----*
//*
//*          1. MERGE PHUS DATA TO RESULT OF MBRLOAD1 BY CAN AND BIC
//*          2. THIS WILL ADD THE BOAN TO PHUS PRELIMINARY TO DOING
//*             MBRADDUP AND MBR SECOND READ - ADD TO CASES IN FINDER
//*          3. OUTPUT FULL FILE FOR ADDUP AND LINKED FILE FOR MBR
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MBRLOAD1.COMB10.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PHUS.COMB10.SSD,DISP=SHR
//LNK DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//*
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO,VOL=(, , ,10)
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPHUS.Y2010.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO,VOL=(, , ,10)
//OUT2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PHUSTEP1.Y2010.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO,VOL=(, , ,10)
//*
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY;
%LET BEGYR=1994;
%LET ENDYR=2010;
%LET ENDMN=12;

/* NOTE: CHANGE THE ENDYR AND ENDMN PARAMETER BEFORE EACH RUN */

/* ASSIGN MACROS VARIABLES TO HANDLE TIME SERIES DATA */

%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&BEGYR %THEN %LET BMN=1; /* START IN JANUARY */
  %ELSE %LET BMN = 1;

```

Appendix A.28
JCL/SAS Code: MRGPHUS1

```
%IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
%ELSE %LET EMN=12;
%DO J=&BMN %TO &EMN;
  %IF &J<10 %THEN %LET MN=0%EVAL(&J);
  %ELSE %LET MN=%EVAL(&J);
  %LET X&K=%EVAL(&YR)%EVAL(&MN);
  %LET K=%EVAL(&K+1);
%END;
%END;
%LET TOT=%EVAL(&K-1);

/* MERGE PHUS ONTO MBRLOAD1 DATA AND OUTPUT MBRADDUP FILE AND LINKED
   FILE TO MERGE ONTO 814 DATA */
DATA OUT1.MBR (DROP=BOAN) OUT2.PHUS (KEEP=SSN BIC PAYD: DIRPAY:
MEDPREM:);
MERGE IN1.MBR (IN=M ) IN2.PHUS (IN=P);
BY CAN BIC;
LENGTH HASPHUS $1;
HASPHUS = 'N';
IF P THEN HASPHUS = 'Y';
IF M THEN DO;
  OUTPUT OUT1.MBR;
  IF SSN NE ' ' THEN OUTPUT OUT2.PHUS;
END;
RUN;
PROC FREQ DATA=OUT1.MBR; TABLES HASPHUS; TITLE 'MERGED PHUS & MBR';
RUN;
PROC PRINT DATA=OUT1.MBR (OBS=20);RUN;
PROC PRINT DATA=OUT2.PHUS (OBS=20);RUN;

%MEND;
%START;
```

Appendix A.29
JCL/SAS Code: MRGPHUS2

```

//$8043MBR JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB10(MRGPHUS2) |
//*          *-----*
//*
//*          1. MERGE PHUS DATA TO RESULT OF MBRLOAD1 BY CAN AND BIC
//*          2. THIS WILL ADD THE BOAN TO PHUS PRELIMINARY TO DOING
//*             MBRADDUP AND MBR SECOND READ - ADD TO CASES IN FINDER
//*          3. OUTPUT FULL FILE FOR ADDUP AND LINKED FILE FOR MBR
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN DD DSN=OPDR.TG.PRD.ETTW.N8043.PHUSTEP1.Y2010.SSD,DISP=SHR,
//          UNIT=TSILO,VOL=SER=(K65167,K13730)
//LNK DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//*
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP3 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP4 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP5 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP6 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.Y2010.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//*
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY;

/*DIVIDE TEMP.PHUS IN ORDER TO SUCCEED IN SORTING IT */
DATA TEMP1.PHUS ;
SET IN.PHUS (OBS=9394094);
RUN;

```

Appendix A.29
JCL/SAS Code: MRGPHUS2

```
DATA TEMP2.PHUS;
  SET IN.PHUS (FIRSTOBS=9394095 OBS=18788189);
RUN;
DATA TEMP3.PHUS;
  SET IN.PHUS (FIRSTOBS=18788190 OBS=30112617);
RUN;

/* SORT PHUS BY SSN AND LINK TO MBR FINDER RECORDS */

PROC SORT DATA=TEMP1.PHUS OUT=TEMP4.PHUS; BY SSN BIC; RUN;
PROC SORT DATA=TEMP2.PHUS OUT=TEMP5.PHUS; BY SSN BIC; RUN;
PROC SORT DATA=TEMP3.PHUS OUT=TEMP6.PHUS; BY SSN BIC; RUN;

PROC SORT DATA=LNK.LINKDI OUT=LINKDI; BY SSN BIC; RUN;
DATA OUT.LINKPHUS;
  MERGE TEMP4.PHUS (IN=P) TEMP5.PHUS (IN=P1) TEMP6.PHUS (IN=P2)
    LINKDI (IN=L);
  BY SSN BIC;
  IF L AND (P OR P1 OR P2);
RUN;

PROC PRINT DATA=OUT.LINKPHUS (OBS=20);TITLE 'FINAL LINKED PHUS';
RUN;
```

Appendix A.30
JCL/SAS Code: DUALPHUS

```
// $8043PHU JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *****
// *
// *      *-----*
// *      | OPDR.TG.PRD.ETTW.N8043.LIB11(DUALPHUS)      |
// *      *-----*
// *
// *      1. OUTPUT SINGLE RECORDS IN SEPARATE FILE
// *      2. PROCESS DUAL RECORDS AND OUTPUT SUMMED DUALS FILE
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// JS030    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
// *
// LNK DD   DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.Y2010.SSD,DISP=SHR
// TEMP DD   DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// OUT1 DD   DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.SING10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// OUT2 DD   DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.DUALS10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
// *
// SYSIN DD *
OPTIONS NOCENTER COMPRESS=YES;

%LET BEGYR=1994;
%LET ENDYR=2010;
%LET ENDMN=12;
%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&BEGYR %THEN %LET BMN=1; /* START IN JANUARY */
  %ELSE %LET BMN = 1;
  %IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
  %ELSE %LET EMN=12;
  %DO J=&BMN %TO &EMN;
    %IF &J<10 %THEN %LET MN=0%EVAL(&J);
    %ELSE %LET MN=%EVAL(&J);
    %LET X&K=%EVAL(&YR)%EVAL(&MN);
    %LET K=%EVAL(&K+1);
  %END;
%END;
%LET TOT=%EVAL(&K-1);
```

Appendix A.30

JCL/SAS Code: DUALPHUS

```

DATA OUT1.LINKPHUS PHUSDUAL;
  /* rename vars wrongly named because macro not updated */
  SET LNK.LINKPHUS
    (RENAME=(DIRPAY01001-DIRPAY01012=DIRPAY1001-DIRPAY1012
              MEDPREM01001-MEDPREM01012=MEDPREM1001-MEDPREM1012
              PAYD01001-PAYD01012=PAYD1001-PAYD1012))
  ;
  BY SSN;
  IF FIRST.SSN AND LAST.SSN THEN OUTPUT OUT1.LINKPHUS;
  ELSE OUTPUT PHUSDUAL;
RUN;
PROC PRINT DATA=PHUSDUAL (OBS=100);TITLE 'BEFORE PROCESSING'; RUN;
DATA OUT2.PHUSDUAL;
  SET PHUSDUAL
    (RENAME=(
      %DO I=1 %TO &TOT;
        PAYD&&X&I=PAYDT&&X&I
        MEDPREM&&X&I = MEDPREMT&&X&I
        DIRPAY&&X&I=DIRPAYT&&X&I
      %END;))
  ;
  BY SSN;
  /* INPUT ARRAYS - RENAMED VARS */
  ARRAY PAYDT(*) %DO I=1 %TO &TOT; PAYDT&&X&I %END;;
  ARRAY PREMT(*) %DO I=1 %TO &TOT; MEDPREMT&&X&I %END;;
  ARRAY DIRT(*) %DO I=1 %TO &TOT; DIRPAYT&&X&I %END;;
  /* OUTPUT ARRAYS - SUMMED PAYMENTS - ORIGINAL NAME */
  ARRAY PAYD(*) %DO I=1 %TO &TOT; PAYD&&X&I %END;;
  ARRAY PREM(*) %DO I=1 %TO &TOT; MEDPREM&&X&I %END;;
  ARRAY DIR(*) %DO I=1 %TO &TOT; DIRPAY&&X&I %END;;

  RETAIN %DO I=1 %TO &TOT; PAYD&&X&I MEDPREM&&X&I DIRPAY&&X&I
    %END;;

DO I = 1 TO DIM(PAYD);
  IF FIRST.SSN THEN DO;
    /* INITIALIZE */
    PAYD(I) = PAYDT(I);
    PREM(I) = PREMT(I);
    DIR(I) = DIRT(I);
  END;
  /* SUM THE AMOUNTS ON SUBSEQUENT RECORDS */
  ELSE DO;
    PAYD(I) + PAYDT(I);
    PREM(I) + PREMT(I);
    DIR(I) + DIRT(I);
  END;
END;

DROP PAYDT: MEDPREMT: DIRPAYT: I;
IF LAST.SSN;

RUN;

```

Appendix A.30
JCL/SAS Code: DUALPHUS

```
PROC PRINT DATA=OUT2.PHUSDUAL (OBS=20);TITLE 'AFTER PROCESSING'; RUN;  
PROC PRINT DATA=OUT1.LINKPHUS (OBS=20);TITLE 'AFTER PROCESSING'; RUN;  
%MEND;  
%START;
```

Appendix A.31
JCL/SAS Code: MBRADDUP

```

//$8043MAD JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(MBRADDUP) |
//*          *-----*
//* MBR RECORDS WERE MERGED WITH PHUS RECORDS TO CREATE MBRPHUS.
//* ON MBRPHUS RECORDS - ADD UP NUMBER OF DEPENDENTS
//* ADD UP BENEFIT AMOUNTS FOR TOTAL DEPENDENT AMOUNT
//* ADD UP TOTAL PAYMENT AMOUNTS FOR DEPENDENTS (PHUS)
//* FILE SORTED BY CAN BIC FOR LINKING TO MBR ORIGINAL RECORDS
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
// SET REG='128M'
//JS010 EXEC SAS9,
//      WORK='200000,100000',           * SPACE IN BLKS *
//      REGION=&REG,
//      PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPHUS.Y2010.SSD1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPHUS.SUMREC.Y2010,
//      DISP=(NEW,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,5)
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO,(VOL=,,,5)
//TEMP2 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO,(VOL=,,,5)
//SYSIN DD *

OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY;
%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE END YEAR AS NEEDED */
%LET ENDMN=12; /* CHANGE END MONTH AS NEEDED */

%macro start;
%let k=1;

%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&BEGYR %THEN %LET BMN=1;
  %ELSE %LET BMN = 1;
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=&BMN %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
  
```

Appendix A.31
JCL/SAS Code: MBRADDUP

```
        %else %let mn=%eval(&j);
        %let x&k=%eval(&yr)%eval(&mn);
        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

data temp1.mbr (KEEP=CAN
                %DO I=1 %TO &TOT;DUEO&&X&I
                PAYO&&X&I
                DPEN&&X&I %END;)
    TEMP2.IDENT (KEEP=CAN SSN BIC);
/* THE FILE IS SORTED WITH THE PRIMARY (BIC=A) FIRST */
/* SUBSEQUENT RECORDS FOR A GIVEN CAN ARE DEPENDENTS */
/* OUTPUT FILE WILL HOLD DEPENDENT COUNTS AND AMOUNTS*/
/* THESE WILL BE ATTACHED TO THE PRIMARIES IN TRF      */
/* IF FIRST BICS ARE NOT = 'A' THEY ARE DACS OR WIDOWS
   AND WILL NOT BE PROCESSED IN THE ADDUP PROCEDURE */

SET IN1.MBR    ; BY CAN BIC;

LENGTH %DO I=1 %TO &TOT; DUEO&&X&I %END; 4;
LENGTH %DO I=1 %TO &TOT; DPEN&&X&I %END; 3;
LENGTH %DO I=1 %TO &TOT; PAYO&&X&I %END; 4;

RETAIN %DO I=1 %TO &TOT; DUEO&&X&I DPEN&&X&I PAYO&&X&I %END;;

ARRAY BEN (*) %DO I=1 %TO &TOT; DUEO&&X&I %END;;
ARRAY PAY (*) %DO I=1 %TO &TOT; PAYO&&X&I %END;;
ARRAY CNT (*) %DO I=1 %TO &TOT; DPEN&&X&I %END;;
ARRAY BPD (*) %DO I=1 %TO &TOT; BPD&&X&I %END;;
ARRAY MBC (*) %DO I=1 %TO &TOT; MBC&&X&I %END;;
ARRAY PD (*) %DO I=1 %TO &TOT; PAYD&&X&I %END;;

IF FIRST.CAN THEN DO;

DO I = 1 TO DIM(BEN);

    /* INITIALIZE THE FIELDS ON THE PRIMARY RECORD */

    BEN(I) = 0; /* SUMMED BENEFIT AMOUNT */
    PAY(I) = 0; /* SUMMED PAYED AMOUNT */
    CNT(I) = 0; /* SUMMED NUMBER OF DEPENDENTS */
END;

/* IF THE FIRST RECORD IS THE ONLY RECORD THE DEPENDENT VARS
   WILL = 0 */

/* THE TEMP2.IDENT FILE WILL HAVE RECORDS ONLY FOR PRIMARIES
   IF THE FIRST BIC IS NOT A PRIMARY - AN ODD OCCURRENCE- THE
   RECORD IS NOT OUTPUT */
```

Appendix A.31
JCL/SAS Code: MBRADDUP

```
IF SUBSTR(BIC,1,1) EQ 'A' THEN OUTPUT TEMP2.IDENT;

END;

ELSE DO; /* ADD UP THE BENEFITS FOR THE AUXILIARIES */

DO I = 1 TO DIM(BEN);
  IF BPD(I) = '1' THEN DO; /* BENEFIT PAID INDICATOR */
    BEN(I) = SUM(BEN(I),MBC(I));
    CNT(I) = SUM(CNT(I),1);
    PAY(I) = SUM(PAY(I),PD(I));
  END;
END;
END;
IF LAST.CAN THEN OUTPUT TEMP1.MBR;
RUN;
PROC PRINT DATA=TEMP2.IDENT (OBS=50); TITLE 'FIRST BICS'; RUN;

DATA OUT.MBR;
MERGE TEMP1.MBR TEMP2.IDENT (IN=I);
BY CAN;
/* ONLY PRIMARY RECORDS OUTPUT IN STEP 1 ARE KEPT AND MERGED WITH
DEPENDENT AMOUNTS */
IF I;
  NOSSN = 0;
  IF SSN = ' ' THEN NOSSN = 1;
RUN;

PROC FREQ DATA=OUT.MBR; TABLES NOSSN ; RUN;

PROC PRINT DATA=OUT.MBR (OBS=50);
  TITLE 'AFTER MERGE ' ;
RUN;

%MEND;

%START;
```

Appendix A.32
JCL: LOAD2JCL

```
//$8043LD2 JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*          RESTART=JS030.SAS9
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(LOAD2JCL) |
//*          *-----*
//* SAS LOAD 5 SECTIONS OF MBR DATA
//* ALL MBR FIELDS LOADED
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS010 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR1.R111011,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR1.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(READ814),
//          DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR2.R111011,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR2.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(READ814),
//          DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR3.R111011,DISP=SHR
```

Appendix A.32
JCL: LOAD2JCL

```
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR3.SSD,
//      DISP=(OLD,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(READ814),
//      DISP=(SHR,PASS,KEEP)
//*
//JS040 EXEC SAS9,
//      WORK='200000,100000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR4.R111011,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR4.SSD,
//      DISP=(OLD,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(READ814),
//      DISP=(SHR,PASS,KEEP)
//*
//JS050 EXEC SAS9,
//      WORK='200000,100000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.MBR5.R111011,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKDI.D1012.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR5.SSD,
//      DISP=(OLD,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(READ814),
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.33
SAS Code: READ814

```
/* *****  
* FILENAME: READ814.SAS  
* PROGRAMMER: MIRIAM LOEWENBERG  
* PURPOSE: TO READ IN THE MBR14 DATA  
* EDITED: OCT 8, 2011  
* THE INPUT FILE IS IN 5 SECTIONS  
* THERE ARE DUAL ELIGIBLES AMONG THE RECORDS AND THERE WILL  
* BE DUPLICATE SSN/BIC COMBINATIONS - THESE WILL BE PROCESSED  
* SEPARATELY AS OUTPUT FROM PROGRAM LOAD2COM  
* *****/  
OPTIONS NOCENTER COMPRESS=BINARY;  
%LET BEGYR=1994;  
%LET ENDYR=2010; /* CHANGE END YEAR AS NEEDED */  
%LET ENDMN=12; /* CHANGE END MONTH AS NEEDED */  
  
%LET MLRECL=11883; /* CHANGE LRECL AS NEEDED */  
  
%MACRO START;  
%LET K=1;  
  
%DO I=&BEGYR %TO &ENDYR;  
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);  
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);  
  %ELSE %LET YR=%EVAL(&I-2000);  
  %IF &I=&BEGYR %THEN %LET BMN=1; /* START IN JANUARY */  
  %ELSE %LET BMN = 1;  
  %IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);  
  %ELSE %LET EMN=12;  
  %DO J=&BMN %TO &EMN;  
    %IF &J<10 %THEN %LET MN=0%EVAL(&J);  
    %ELSE %LET MN=%EVAL(&J);  
    %LET X&K=%EVAL(&YR)%EVAL(&MN);  
    %LET K=%EVAL(&K+1);  
  %END;  
%END;  
%LET TOT=%EVAL(&K-1);  
  
DATA TEMP.MBR;  
INFILE IN1 TRUNCOVER;  
INPUT  
  @1 CAN $9.  
  @10 TAC $1. /* TYPE OF AWARD CODE */  
  @11 RP $1. /* RACE OF PRIMARY */  
  @12 LSPA 5.2 /*LUMP SUM PAYMENT AMOUNT */  
  @19 PNOB $2. /* NUMBER OF BENES IN PAYMENT */  
  @29 DOC $3. /* DISTRICT OFFICE CODE */  
  @32 STATE $2.  
  @34 COUNTY $3.  
  @42 LSSC_DSALLW $1. /* LUMP SUM STATUS CODE -DISALLOWED */  
  @43 LSSC_FUNR $1. /* LUMP SUM STAT CODE -FUNERAL HOME */  
  @44 LSSC_BURL $1. /* LUMP SUM STAT CODE -BURIAL EXP */  
  @45 LSSC_LVSPS $1. /* LUMP SUM STAT CODE -LIVING W/SPOUSE*/  
  @46 LSSC_AUPTMT $1. /* LUMP SUM STAT CODE -AUTHORIZED PYMT*/
```

Appendix A.33
SAS Code: READ814

```

@47  LSSC_WID          $1. /* LUMP SUM STAT CODE -ENTITLED WIDOW*/
@48  LSSC_CHLD        $1. /* LUMP SUM STAT CODE -ENTITLED CHILD*/
@49  LSFDX            $6. /* LUMP SUM FILING DATE */
@55  LSAP             5.2 /* LUMP SUM AWARD AMOUNT */
@60  LSDC             $1. /* LUMP SUM DISALLOWANCE CODE */
@61  BIC              $2. /* BENE IDENTIFICATION CODE */
@63  LAF              $2. /* LEDGER ACCOUNT FILE */
@65  DOB ??MMDDYY8. /* DATE OF BIRTH */
@73  BGN              $15. /* BENE GIVEN NAME */
@88  BLN              $20. /* BENE LAST NAME */
@108 DOEIX           $6. /* EARLIEST MONTH ENTITLED TO BENEFITS */
@114 DOECX           $6. /* DATE OF CURRENT ENTITLEMENT */
@120 CEC              $1. /* CURRENT ENTITLEMENT CODE */
@129 DOSTX           $6. /* DATE OF SUSPENSION/TERMINATION */
@135 SEX              $1. /* GENDER */
@136 RELATC          $1. /* RELATIONSHIP CODE */
@137 MBP             5.1 /* MONTHLY BENEFIT PAID */
@142 LANG             $1. /* WRITTEN LANGUAGE */
@143 TOC_NUM         $2. /* NUMBER OF TOC OCCURRENCES */
@;
POS=145;
%DO I=1 %TO 20;
  INPUT
  @POS TOC&I $1. /* TYPE OF CLAIM */
  @;
  POS=POS+1;
%END;
POS=165;
%DO I=1 %TO 20;
  INPUT
  @POS TOC_STARX&I $6. /* TOC START DATE */
  @;
  POS=POS+6;
%END;
INPUT @285 RDD          $3. /* REASON FOR DISALLOWANCE/DENIAL */
@288 TOP              $1. /* TYPE OF PAYEE */
@289 CC               $1. /* TYPE OF PAYEE */
@290 GS              $1. /* GUARDIAN STATUS */
@291 RZIP            $5. /* RESIDENCE ZIP CODE */
@296 SIFT            $1. /* SECURITY INCOME FILE TYPE */
@297 SIEDX           $6. /* SSI TERMINATION DATE */
@303 SLAC            $1. /* SSI LIVING ARRANG CODE */
@304 SISC            $1. /* SSI INCOME STATUS CODE */
@305 BOAN            $9. /* BENE OWN ACCOUNT NUMBER */
@314 BDOD ??MMDDYY8. /* DATE OF DEATH */
@322 NDOF            2. /* NUM OCCURRENCES DATE IF FILING */
@;
POS=324;
%DO I=1 %TO 25;
  INPUT
  @POS BDOF&I ??MMDDYY8. /* BENEFICIARY DATE OF FILING */
  @;
  POS=POS+8;

```

Appendix A.33
SAS Code: READ814

```
%END;

POS=524;
%DO I=1 %TO 25;
  INPUT
  @POS BDOE_START&I $6. /* BDOE START DATE */
  @;
  POS=POS+6;
%END;

POS=674;
%DO I=1 %TO 25;
  INPUT
  @POS BDOE_TERM&I $6. /* BDOE TERM DATE */
  @;
  POS=POS+6;
%END;

POS=824;
%DO I=1 %TO 25;
  INPUT
  @POS HBIC&I $2./* HISTORIC BIC */
  @;
  POS=POS+2;
%END;

POS=874;
%DO I=1 %TO 25;
  INPUT
  @POS HCEC&I $1./* HISTORIC CEC */
  @;
  POS=POS+1;
%END;

INPUT @899 HI_STRTX $6. /* HI ENROLLMENT START DATE */

@905 HI_TERM $6. /* DATE OF HI TERMINATION */
@911 SMI_STAX $6. /* SMI ENROLLMENT START DATE */
@917 SMI_TERX $6. /* DATE OF SMI TERMINATION */
@923 NPIA 3.
@;

IF SUBSTR(BIC,1,1) IN ('A') THEN SSN=CAN;
ELSE SSN=BOAN;

POS=926;
%DO I=1 %TO 50;
  INPUT
  @POS PIEDX&I $6. /* PIA EFFECTIVE DATE */
  @;
  POS=POS+6;
%END;
```

Appendix A.33
SAS Code: READ814

```
POS=1826;
%DO I=1 %TO 50;
  INPUT
  @POS PIA&I      5.1 /* PIA AMOUNT */
  @;
  POS=POS+5;
%END;

POS=2576;
%DO I=1 %TO 50;
  INPUT
  @POS IME&I      4. /* AVERAGE INDEXED MONTHLY EARNINGS */
  @;
  POS=POS+4;
%END;

POS=3176;
%DO I=1 %TO 50;
  INPUT
  @POS PIARFC&I  $1. /* REASON FOR CHANGE IN PIA */
  @;
  POS=POS+1;
%END;

INPUT @4676 DLMX $6. /* DATE LAST INSURED FOR DISABILITY */
      @4682 NODF 2.
      @;
POS=4684;
%DO I=1 %TO 12;
  INPUT
  @POS      DDO&I   MMDDYY8. /* DATE OF DISABILITY ONSET */
  @;
  POS=POS+8;
%END;

POS=4780;
%DO I=1 %TO 12;
  INPUT
  @POS      DAC&I   $1. /* DISABILITY AWARD CODE */
  @;
  POS=POS+1;
%END;

POS=4792;
%DO I=1 %TO 12;
  INPUT
  @POS      LOD&I   $1. /* LEVEL OF DENIAL CODE */
  @;
  POS=POS+1;
%END;

POS=4804;
%DO I=1 %TO 12;
```

Appendix A.33
SAS Code: READ814

```
INPUT
@POS    DOEDX&I    $6. /* DATE OF ENTITLEMENT TO DIB */
@;
POS=POS+6;
%END;

POS=4876;
%DO I=1 %TO 12;
INPUT
@POS    T2_DOEDX&I    $6. /* T2 DATE OF ENTITLEMENT TO DIB */
@;
POS=POS+6;
%END;

POS=4948;
%DO I=1 %TO 12;
INPUT
@POS    DDBCX&I    $6. /* DATE OF DISAB BENEFIT CESSATION */
@;
POS=POS+6;
%END;

POS=5020;
%DO I=1 %TO 12;
INPUT
@POS    DSDX&I    $6. /* DISAB ADJUDICATION DATE */
@;
POS=POS+6;
%END;

POS=5092;
%DO I=1 %TO 12;
INPUT
@POS    HDDX&I    $6. /* HEARING DECISION DATE */
@;
POS=POS+6;
%END;

POS=5164;
%DO I=1 %TO 12;
INPUT
@POS    SDSX&I    $6. /* SGA DISAB CESSATION */
@;
POS=POS+6;
%END;

POS=5236;
%DO I=1 %TO 12;
INPUT
@POS    ADCX&I    $6. /* APPLICANTS DISAB CESSATION */
@;
POS=POS+6;
%END;
```

Appendix A.33
SAS Code: READ814

```
POS=5308;
%DO I=1 %TO 12;
  INPUT
  @POS    APSX&I    $6. /* APPEALS */
  @;
  POS=POS+6;
%END;

POS=5380;
%DO I=1 %TO 12;
  INPUT
  @POS    EBDX&I    $6. /* EPE BEGIN DATE */
  @;
  POS=POS+2;
%END;

POS=5452;
%DO I=1 %TO 12;
  INPUT
  @POS    BDC&I    $2. /* BASIS FOR DENIAL CODE */
  @;
  POS=POS+2;
%END;

POS=5476;
%DO I=1 %TO 12;
  INPUT
  @POS    CDR&I    $1. /* CESSATION DISAB REASON */
  @;
  POS=POS+1;
%END;

POS=5488;
%DO I=1 %TO 12;
  INPUT
  @POS    CSA&I    $1. /* CURRENT SGA */
  @;
  POS=POS+1;
%END;

POS=5524;
%DO I=1 %TO 12;
  INPUT
  @POS    DIG&I    $10. /* PRIMARY DIAGNOSIS */
  @;
  POS=POS+10;
%END;

POS=5644;
%DO I=1 %TO 12;
  INPUT
  @POS    SDIG&I    $10. /* SECONDARY DIAGNOSIS */
```

Appendix A.33
SAS Code: READ814

```
@;  
  POS=POS+10;  
%END;  
  
POS=5764;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS MBA&&X&I      5.1 /* MONTHLY BENEFIT AMOUNT */  
  @;  
  POS=POS+5;  
%END;  
  
POS=6784;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS MBC&&X&I      5.1 /* MONTHLY BENEFIT CREDITED */  
  @;  
  POS=POS+5;  
%END;  
  
POS=7804;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS MBP&&X&I      5.1 /* MONTHLY BENEFIT PAID */  
  @;  
  POS=POS+5;  
%END;  
  
POS=9640;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS BPD&&X&I      $1. /* BENEFIT PAID DESIGNATION */  
  @;  
  POS=POS+1;  
%END;  
  
POS=9844;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS RFD&&X&I      $1. /* REASON FOR DEDUCTION */  
  @;  
  POS=POS+1;  
%END;  
  
POS=10048;  
%DO I=1 %TO &TOT;  
  INPUT  
  @POS WIC&&X&I      $1. /* WORK INDICATION CODE */  
  @;  
  POS=POS+1;  
%END;  
  
POS=10252;  
%DO I=1 %TO &TOT;
```

Appendix A.33

SAS Code: READ814

```

INPUT
@POS RFST&&X&I          $6. /* REASON FOR SUSPENSION */
@;
POS=POS+6;
%END;

POS=11476;
%DO I=1 %TO &TOT;
  INPUT
  @POS LAF&&X&I          $2. /* MONTHLY LAF STATUS */
  @;
  POS=POS+2;
%END;
;
/* CHANGE THE MBR14 DATES FROM CHARACTER TO NUMERIC */

ARRAY OLD LSFDX DOEDX1-DOEDX12 SIEDX
      DDBCX1-DDBCX12 DSDX1-DSDX12 HDDX1-HDDX12 SDSX1-SDSX12
      ADCX1-ADCX12 APSX1-APSX12 EBDX1-EBDX12
      DOEIX DOECX
      DOSTX HI_STRTX HI_TERMX PIEDX1-PIEDX50
      SMI_STAX SMI_TERX
      TOC_STARX1-TOC_STARX20 DLMX T2_DOEDX1-T2_DOEDX12;

ARRAY NEW LSFDF ENTDAT1-ENTDAT12 SIED
      DDBC1-DDBC12 DSD1-DSD12 HDD1-HDD12 SDS1-SDS12
      ADC1-ADC12 APS1-APS12 EBD1-EBD12
      DOEI DOEC
      DOST HI_START HI_TERM PIED1-PIED50
      SMI_STAR SMI_TERM
      TOC_START1-TOC_START20 DLM T2_DOED1-T2_DOED12;

DO I=1 TO DIM(OLD);
  NEW(I)=INPUT('01' || OLD(I), ??DDMMYY8.);
END;

DROP LSFDF DOEDX1-DOEDX12 SIEDX
      DOEIX DOECX I POS
      DDBCX1-DDBCX12 DSDX1-DSDX12 HDDX1-HDDX12 SDSX1-SDSX12
      ADCX1-ADCX12 APSX1-APSX12 EBDX1-EBDX12
      DOSTX HI_STRTX HI_TERMX PIEDX1-PIEDX50
      SMI_STAX SMI_TERX
      TOC_STARX1-TOC_STARX20 DLMX T2_DOEDX1-T2_DOEDX12;

LENGTH BDOD DOB DOEC DOEI DOST HI_START HI_TERM SIED
      SMI_STAR SMI_TERM MBP LSFDF TOC_START1-TOC_START20 DLM
      T2_DOED1-T2_DOED12
%DO I=1 %TO 12;
  DDO&I
  ENTDAT&I
  DDBC&I DSD&I HDD&I SDS&I EBD&I
  ADC&I APS&I
%END;

```

Appendix A.33
SAS Code: READ814

```
%DO I=1 %TO 50;
  IME&I
  PIA&I
  PIED&I
%END;
%DO I=1 %TO 15;
  BDOF&I
%END;
%DO I=1 %TO &TOT;
  MBA&&X&I
  MBC&&X&I
  MBP&&X&I
%END;
4;
RUN;

PROC SORT DATA=TEMP.MBR OUT=TEMP1.MBR; BY SSN BIC; RUN;
/* SSN STAFF INDICATED THAT 2% OF RECORDS MAY HAVE THE WRONG
BOANS */
PROC SORT DATA=IN2.LINKDI NODUPKEY OUT=LINKDI; BY SSN BIC; RUN;

DATA OUT.MBR;
MERGE TEMP1.MBR (IN=M) LINKDI (KEEP=SSN BIC IN=LINK);
BY SSN BIC;
IF M AND LINK;
RUN;

PROC CONTENTS DATA=OUT.MBR; RUN;

/* CHANGE FOR TRF07 - WE ARE GOING TO PROCESS THE DUAL ELIGIBLES
AFTER THE SEGMENTS HAVE BEEN COMBINED. THERE WILL BE DUPLICATE
SSN/BIC COMBINATIONS IN EACH SEGMENT. THESE DUPLICATE RECORDS
WILL BE PROCESSED SEPARATELY AND THE INFORMATION COMBINED INTO
1 RECORD. */

PROC FREQ DATA=OUT.MBR ;TABLES BIC HBIC: CEC HCEC: TOC: RZIP LAF;
RUN;
PROC PRINT DATA=OUT.MBR (OBS=25 );
FORMAT ENTDAT1-ENTDAT12 LSF D BDOF1-BDOF15 DOB BDOD DD01-DD012
DDBC1-DDBC12 DSD1-DSD12 HDD1-HDD12 SDS1-SDS12
ADC1-ADC12 APS1-APS12 EBD1-EBD12
DOEI DOEC SIED
DOST PIED1-PIED50 DLM TOC_START1-TOC_START20 T2_DOED1-
T2_DOED12
SMI_STAR SMI_TERM YMMDD10.;
RUN;

%MEND;
%START;
```

Appendix A.34

JCL/SAS Code: LOAD2COM

```

//$8043LD2 JOB (12510000,T715,,SAS,,ITC9FL),8043MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(LOAD2COM) |
//*          *-----*
//*
//*          1.  COMBINE EACH SEGMENT OF MBR LOAD2 FILE (READ814)
//*          2.  OUTPUT SINGLE RECORDS IN 1 FILE AND DUAL ELIG IN ANOTHER
//*          2.  SORTED BY SSN-BIC
//*          3.  READ814 SEGMENTS WERE SORTED BY SSN/BIC
//*
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS030    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1 DD   DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR1.SSD,DISP=SHR
//IN2 DD   DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR2.SSD,DISP=SHR
//IN3 DD   DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR3.SSD,DISP=SHR
//IN4 DD   DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR4.SSD,DISP=SHR
//IN5 DD   DSN=OPDR.TG.PRD.ETTW.N8043.Y2011.MBR5.SSD,DISP=SHR
//OUT1 DD   DSN=OPDR.TG.PRD.ETTW.N8043.READ814.SINGS10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO,VOL=(, , , 3)
//OUT2 DD   DSN=OPDR.TG.PRD.ETTW.N8043.READ814.DUALS10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(, , , 10)
//SYSIN DD   *

OPTIONS NOCENTER COMPRESS=YES;
DATA OUT1.MBR OUT2.DUALS;
  SET IN1.MBR
      IN2.MBR
      IN3.MBR
      IN4.MBR
      IN5.MBR;
  BY SSN BIC;
  IF SSN = ' ' THEN DELETE;
  IF FIRST.SSN AND LAST.SSN THEN OUTPUT OUT1.MBR;
  ELSE OUTPUT OUT2.DUALS;
RUN;
PROC PRINT DATA=OUT1.MBR (OBS=25); TITLE 'SINGLE RECORDS';RUN;
PROC PRINT DATA=OUT2.DUALS (OBS=25); TITLE 'DUAL RECORDS'; RUN;
PROC FREQ DATA=OUT1.MBR; TABLES BIC TOC; RUN;

```

Appendix A.35

JCL/SAS Code: PROCDUAL

```

//$8043DUA JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*          RESTART=JS020.SAS9
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(PROCDUAL) |
//*          *-----*
//* PROCESS DUAL ELIGIBLE RECORDS
//* ALL MBR FIELDS LOADED
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS010 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
//*
//IN DD DSN=OPDR.TG.PRD.ETTW.N8043.READ814.DUALS10.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP3 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.PROCDUAL.DUALS10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY;
%LET BEGYR=1994;
%LET ENDYR=2010;
%LET ENDMN=12;
%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&BEGYR %THEN %LET BMN=1; /* START IN JANUARY */
  %ELSE %LET BMN = 1;
  %IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
  %ELSE %LET EMN=12;
  %DO J=&BMN %TO &EMN;
    %IF &J<10 %THEN %LET MN=0%EVAL(&J);
    %ELSE %LET MN=%EVAL(&J);
    %LET X&K=%EVAL(&YR)%EVAL(&MN);
  %END;
%END;

```

Appendix A.35
JCL/SAS Code: PROCDUAL

```
        %LET K=%EVAL(&K+1);
    %END;
%END;
%LET TOT=%EVAL(&K-1);

PROC PRINT DATA=IN.DUALS (OBS=20); TITLE 'BEFORE PROCESSING';
VAR SSN CAN BIC NODF DOEI DOEC  ENTDAT1-ENTDAT3 PIED1 MBC: BPD;;
RUN;

/* DELETE CASES WITH NO DISABILITY DATA.
   PUT ASIDE NEWLY CREATED SINGLE RECORDS AFTER DELETIONS
   ADD THESE TO THE FINAL PROCESSED FILE */
DATA TEMP.DUALS;
SET IN.DUALS;
BY SSN BIC;
IF NODF = 0 THEN DELETE;
RUN;

DATA TEMP1.FINAL SINGLE ;
SET TEMP.DUALS ;
BY SSN BIC;
IF FIRST.SSN AND LAST.SSN THEN OUTPUT SINGLE;
ELSE OUTPUT TEMP1.FINAL;
RUN;

/* TAKE THE MOST RECENT RECORD (LATEST DOEC) - THE OTHER RECORDS ARE
   CONSIDERED SUPPLEMENTARY. CERTAIN VARIABLES ARE RETAINED
   FROM THE SUPPLEMENTARY RECORDS AND RENAMED ENDING IN D.
   FOR THE CASES WHERE DOEC IS THE SAME ON BOTH RECORDS USE THE
   MOST RECENT (LATEST) FIRST ENTDATE.
   FOR THE CASES WHERE ENTDATE IS THE SAME ON BOTH RECORDS USE THE
   RECORD WITH LATEST PIED1.
   FOR THE REMAINDER SELECT THE RECORD WITH THE BIC = A.
   FOR THE FEW RECORDS REMAINING SORT BY BIC, PROCESS THE BENEFIT
   AMOUNTS, AND OUTPUT THE LAST RECORD.
   */

DATA TEMP1.FINAL FLAGDATA (KEEP=SSN DOECFLAG ENTFLAG PIEDFLAG
BICAFLAG);
SET TEMP1.FINAL;
BY SSN BIC;

/* GET THE FIRST ENTDATE - SOMETIMES THE FIRST IS BLANK AND A
   SUCCESSION ONE IS POPULATED */
FIRSTENT = .;
ARRAY ENT(*) ENTDAT1-ENTDAT12;
DO I = 1 TO DIM(ENT) UNTIL (ENT(I) GT .Z);
    FIRSTENT = ENT(I);
END;

/* CREATE VARIABLE FOR GROUPING BIC CODES AND SORTING LATER */
BICNDEX = 0;
IF SUBSTR(BIC,1,1) = 'A' THEN BICNDEX = 1;
```

Appendix A.35
JCL/SAS Code: PROCDUAL

```
/* RETAIN TEMPORARY VARIABLES FOR COMPARISONS OF 1ST TO LAST RECORD
*/
RETAIN CURRELIG ENT1 PIEDD BICC;

/* CREATE FLAGS TO SHOW WHEN DATES ARE NOT EQUAL ON ALL RECORDS */
ARRAY FLGS(*) DOECFLAG ENTFLAG PIEDFLAG BICAFLAG;

IF FIRST.SSN THEN DO;
  CURRELIG = DOEC;
  ENT1 = FIRSTENT;
  PIEDD = PIED1;
  BICC = SUBSTR(BIC,1,1);
  DO I = 1 TO DIM(FLGS);
    FLGS(I) = 0;
  END;
END;
/* CHECK ALL SUCCESSIVE RECORDS - COMPARISON FIELDS ARE RETAINED
*/
ELSE DO;
  IF CURRELIG NE DOEC THEN DOECFLAG = 1;
  ELSE IF ENT1 NE FIRSTENT THEN ENTFLAG = 1;
  ELSE IF PIEDD NE PIED1 THEN PIEDFLAG = 1;
  ELSE IF BICC = 'A' OR SUBSTR(BIC,1,1) = 'A' THEN BICAFLAG = 1;
END;
/* OUTPUT THE RECORDS */
IF LAST.SSN THEN DO;

  OUTPUT FLAGDATA;

  END; /* LAST SSN */

OUTPUT TEMP1.FINAL;

RUN;

/* MERGE THE FLAGS BACK ONTO THE DATA */

DATA TEMP3.DIFFELIG DIFFENT DIFFPIED DIFFA SAMEALL;
MERGE TEMP1.FINAL
  (DROP=DOECFLAG ENTFLAG PIEDFLAG BICAFLAG) FLAGDATA;
BY SSN;

/* USE THE LOGICAL SEQUENCE NOTED ABOVE TO OUTPUT SEPARATE DATA
SETS*/

IF DOECFLAG THEN OUTPUT TEMP3.DIFFELIG;
ELSE IF ENTFLAG THEN OUTPUT DIFFENT;
ELSE IF PIEDFLAG THEN OUTPUT DIFFPIED;
ELSE IF BICAFLAG THEN OUTPUT DIFFA;
ELSE OUTPUT SAMEALL;
RUN;
```


Appendix A.35
JCL/SAS Code: PROCDUAL

```
DO I = 1 TO &TOT;
  IF FIRST.SSN THEN DO;
    MBAT(I) = 0;
    MBCT(I) = 0;
    MBPT(I) = 0;
    BPDT(I) = '0';

    /* THESE TEMPORARY VARS WILL BE COMPUTED ONLY IF BPD = 1 ON EITHER
    OR BOTH RECORDS.
    IF BPD IS NEVER = 1 ON ANY OF THE RECORDS WE
    STILL WANT TO RETAIN THE VALUES ON THE LAST RECORD ACCORDING TO
    THE SORT - AMOUNT VALUES ARE ON THE RECORDS EVEN IF PAYMENTS
    ARE IN SUSPENSION - SEE PROCESSING FOR LAST.SSN */

    IF BPDA(I) = '1' THEN DO;
      BPDT(I) = '1';
      MBAT(I) = MBAA(I);
      MBCT(I) = MBCA(I);
      MBPT(I) = MBPA(I);
    END;
  END;
  ELSE IF NOT FIRST.SSN THEN DO; /* SUBSEQUENT RECORDS */
    /* THIS WILL SUM WITH WHAT IS ON THE PREVIOUS RECORD */
    IF BPDA(I) = '1' THEN DO;
      BPDT(I) = '1';
      MBAT(I) = SUM(MBAT(I),MBAA(I));
      MBCT(I) = SUM(MBCT(I),MBCA(I));
      MBPT(I) = SUM(MBPT(I),MBPA(I));
    END;
  END;

END; /* WHOLE ARRAY */

/* ARRAYS FOR THE DISABILITY FIELDS - WE CHOOSE THE LAST POPULATED
*/
ARRAY DDO (*) DDO1-DDO12;
ARRAY DAC ($) $ DAC1-DAC12;
ARRAY LOD ($) $ LOD1-LOD12;
ARRAY ENT (*) ENTDAT1-ENTDAT12;
ARRAY DDBC (*) DDBC1-DDBC12;
ARRAY DSD (*) DSD1-DSD12;
ARRAY HDD (*) HDD1-HDD12;
ARRAY SDS (*) SDS1-SDS12;
ARRAY ADC (*) ADC1-ADC12;
ARRAY APS (*) APS1-APS12;
ARRAY EBD (*) EBD1-EBD12;
ARRAY BDC ($) $ BDC1-BDC12;
ARRAY CDR ($) $ CDR1-CDR12;
ARRAY CSA ($) $ CSA1-CSA12;
ARRAY DIG ($) $ DIG1-DIG12;
ARRAY SDIG (*) $ SDIG1-SDIG12;

/* THE FOLLOWING MACROS CREATE THE NEW VARIABLE WE ARE GOING TO
```

Appendix A.35

JCL/SAS Code: PROCDUAL

KEEP FROM THE (PREVIOUS) SUPPLEMENTARY RECORD ON THE KEPT RECORD.
THE DATA IS SORTED BY THE VARIABLE WHOSE LATEST VALUE WE
HAVE CHOSEN AS FLAGGING THE RECORD WE WANT TO RETAIN.
SINCE WE RETAIN THE NEW VARIABLE IN PROCESSING ALL THE RECORDS
EXCEPT THE LAST ONE, THE PREVIOUS RETAINED VALUE WILL BE KEPT
ON THE LAST (KEPT) RECORD. EACH DATA SET OUTPUT USING THE FLAGS
ABOVE IS PROCESSED SEPARATELY */

```
%MACRO LASTIT (ARR,VAR); /* FOR NUMERIC VARIABLES */
```

```
/* INTITALIZE THE VARIABLES BECAUSE THEY ARE ALL MISSING ON MANY
RECORDS */
RETAIN &VAR;
IF FIRST.SSN THEN &VAR = .;
IF NOT LAST.SSN THEN DO;
DO I = 1 TO DIM(&ARR);
/* WE HAVE TO LOOP THRU ALL VARS BECAUSE EARLIER VAR MAY BE
MISSING
AND LATER VAR POPULATED */
IF &ARR(I) GT .Z THEN &VAR = &ARR(I);
END;
END;
```

```
%MEND LASTIT;
%LASTIT(DDO,DDODT);
%LASTIT(ENT,ENTDATDT);
%LASTIT(DDBC,DBBCDT);
%LASTIT(DSD,DSDDT);
%LASTIT(HDD,HDDDT);
%LASTIT(SDS,SDSDT);
%LASTIT(ADC,ADCDT);
%LASTIT(APS,APSDT);
%LASTIT(EBD,EBDDT);
```

```
%MACRO LASTIT1(ARR,VAR); /* FOR CHARACTER VARIABLES */
LENGTH BDCDT $2 DIGDT SDIGDT $4;
```

```
/* INTITALIZE THE VARIABLES BECAUSE THEY ARE ALL MISSING ON MANY
RECORDS */
RETAIN &VAR;
IF FIRST.SSN THEN &VAR = ' ';
IF NOT LAST.SSN THEN DO;
DO I = 1 TO DIM(&ARR);
IF &ARR(I) NE ' ' THEN &VAR = &ARR(I);
END;
END;
```

```
%MEND LASTIT1;
%LASTIT1(DAC,DACDT);
%LASTIT1(LOD,LODDT);
%LASTIT1(BDC,BDCDT);
%LASTIT1(CDR,CDRDT);
%LASTIT1(CSA,CSADT);
```

Appendix A.35
JCL/SAS Code: PROCDUAL

```
%LASTIT1(DIG,DIGDT);  
%LASTIT1(SDIG,SDIGDT);
```

```
/* NOW PROCESS THE CHOSEN RECORD */  
IF LAST.SSN THEN DO;  
/* TAKE THE OUTCOMES OF THE AMOUNT VARS AND BPD.  
IF THE AMOUNTS HAVE BEEN SUMMED INTO THE TEMP VARS THEN TAKE  
THE SUMMED VALUE AS THE FINAL AMOUNT. ELSE TAKE THE ORIGINAL  
AMOUNT ON THE RECORD WHICH BECOMES THE LAST RECORD IN THE SORT
```

*/

```
DO I = 1 TO &TOT;  
BPD(I) = BPDT(I);  
IF BPDT(I) = ' ' THEN BPD(I) = BPDA(I);  
IF MBAT(I) GT 0 THEN MBA(I) = MBAT(I);  
ELSE MBA(I) = MBAA(I);  
IF MBCT(I) GT 0 THEN MBC(I) = MBCT(I);  
ELSE MBC(I) = MBCA(I);  
IF MBPT(I) GT 0 THEN MBPO(I) = MBPT(I);  
ELSE MBPO(I) = MBPA(I);  
END;
```

```
/* TAKE RETAINED PREVIOUS VALUE FOR SUPPLEMENTARY VARIABLES */  
LENGTH BDCD $2 DIGD SDIGD $4;  
DOECD = DOECT;  
DOEID = DOEIT;  
DDOD = DDODT;  
DACD = DACDT;  
LODD = LODDT;  
ENTDATD = ENTDATDT;  
DDBCDCD = DDBCDCDT;  
DSDD = DSDDT;  
HDDD = HDDDTC;  
SDSD = SDSDT;  
ADCD = ADCDT;  
APSD = APSDT;  
EBDD = EBDDT;  
BDCD = BDCDT;  
CDRD = CDRDT;  
CSAD = CSADT;  
DIGD = DIGDT;  
SDIGD = SDIGDT;
```

```
DROP DOECFLAG ENTFLAG PIEDFLAG BICAFLAG I  
MBAA: MBCA: MBPA: BPDA:  
MBAT: MBCT: MBPT: BPDT:  
PIEDD  
CURRELIG  
ENT1 FIRSTENT  
DOECT  
DOEIT  
DDODT  
DACDT
```

Appendix A.35
JCL/SAS Code: PROCDUAL

```
LODDT
ENTDATDT
DDBCDT
DSDDT
HDDDT
SDSDT
ADCDDT
APSDT
EBDDT
BDCDDT
CDRDT
CSADT
DIGDT
SDIGDT
BICNDEX;

OUTPUT;

END;

RUN;

TITLE "AFTER PROCESSING - &FILE RECORDS";
PROC PRINT DATA=&FILE (OBS=20); VAR SSN BIC DOEC: DOEI: ENTDAT1-
ENTDAT3
    ENTDATD DDO1-DDO3 DDOD DAC1-DAC3 DACD LOD1-LOD3 LODD MBC: BPD;;
RUN;

%MEND CHOOSIT;

%CHOOSIT(TEMP3.DIFFELIG,DOEC);
%CHOOSIT(DIFFENT,FIRSTENT);
%CHOOSIT(DIFFPIED,PIED1);
%CHOOSIT(DIFFA,BICNDEX);
%CHOOSIT(SAMEALL,BIC);

/* COMBINE ALL FILES FOR FINAL DATASET */
DATA OUT.DUALS;
    SET SINGLE TEMP3.DIFFELIG DIFFENT DIFFPIED DIFFA SAMEALL;
    BY SSN;

RUN;
TITLE 'FINAL SELECTED RECORDS';
PROC CONTENTS DATA=OUT.DUALS; RUN;
PROC PRINT DATA=OUT.DUALS (OBS=20); RUN;
%MEND;
%START;
```

Appendix A.36
JCL/SAS Code: MRGDUALS

```

//$8043MRG  JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
//*          RESTART=JS020.SAS9
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB11(MRGDUALS) |
//*          *-----*
//* PROCESS DUAL ELIGIBLE RECORDS
//* ALL MBR FIELDS LOADED
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*****
//*
//JS010     EXEC SAS9,
//          WORK='200000,100000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.READ814.SINGS10.SSD,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.PROCDUAL.DUALS10.SSD,DISP=SHR
//IN3      DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.SING10.SSD,DISP=SHR
//IN4      DD DSN=OPDR.TG.PRD.ETTW.N8043.LINKPHUS.DUALS10.SSD,DISP=SHR
//TEMPS    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//TEMPD    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(2000,1000),RLSE)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.N8043.READ814.COMB10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO,VOL=(,,,2)
//SYSIN    DD *
OPTIONS NOCENTER COMPRESS=YES;
DATA TEMPS.SING10;
MERGE IN1.MBR (IN=M) IN3.LINKPHUS ;
BY SSN;
IF NOT M THEN DELETE;

DROP I;

RUN;
PROC PRINT DATA=TEMPS.SING10 (OBS=10);TITLE 'MERGED SINGLES'; RUN;
DATA TEMPD.DUAL10;
MERGE IN2.DUALS (IN=D) IN4.PHUSDUAL (IN=DU);
BY SSN;
IF NOT D THEN DELETE;
/* BUILD DUAL ELIGIBLE FLAG */
DUALELIG = 1;

DROP BICC I;

RUN;
PROC PRINT DATA=TEMPD.DUAL10 (OBS=10);TITLE 'MERGED DUALS'; RUN;

```

Appendix A.36
JCL/SAS Code: MRGDUALS

```
DATA OUT.MBR;
MERGE TEMPS.SING10 TEMPD.DUAL10 (IN=DU);
  BY SSN;

  /* SET DUALELIG FLAG TO 0 FOR SINGLES */
  IF DUALELIG = . THEN DUALELIG = 0;

LABEL
DUALELIG = 'FLAG FOR DUAL ELIGIBLE BENEFICIARY'
DDOD = 'DUAL ELIG-DATE OF DISABILITY ONSET-PREVIOUS RECORD'
DACD = 'DUAL ELIG-DISABILITY AWARD CODE-PREVIOUS RECORD'
LODD = 'DUAL ELIG-LEVEL OF DENIAL CODE-PREVIOUS RECORD'
ENTDATD = 'DUAL ELIG-DATE OF ENTITLEMENT-PREVIOUS RECORD'
DDBCD = 'DUAL ELIG-DATE OF BENEFIT CESSATION-PREVIOUS RECORD'
DSDD = 'DUAL ELIG-ADJUDICATION DATE-PREVIOUS RECORD'
HDDD = 'DUAL ELIG-HEARING DECISION DATE-PREVIOUS RECORD'
SDSD = 'DUAL ELIG-SGA DISABILITY CESSATION DATE-PREVIOUS RECORD'
ADCD = 'DUAL ELIG-APPLICANT DISAB CESSATION DATE-PREVIOUS RECORD'
APSD = 'DUAL ELIG-APPEALS DATE-PREVIOUS RECORD'
EBDD = 'DUAL ELIG-EPE BEGIN DATE-PREVIOUS RECORD'
BDCD = 'DUAL ELIG-BASIS FOR DENIAL CODE-PREVIOUS RECORD'
CDRD = 'DUAL ELIG-DISABILITY CESSATION REASON-PREVIOUS RECORD'
CSAD = 'DUAL ELIG-CURRENT SGA STATUS-PREVIOUS RECORD'
DIGD = 'DUAL ELIG-PRIMARY DIAGNOSIS-PREVIOUS RECORD'
SDIGD = 'DUAL ELIG-SECONDARY DIAGNOSIS-PREVIOUS RECORD'
DOEID = 'DUAL ELIG-DATE OF INITIAL ELIG-PREVIOUS RECORD'
DOECD = 'DUAL ELIG-DATE OF CURRENT ELIG-PREVIOUS RECORD'
;
RUN;
TITLE 'FINAL MERGED MBR DATA WITH DUALS AND PHUS';
PROC FREQ DATA=OUT.MBR; TABLES DUALELIG; RUN;
PROC CONTENTS DATA=OUT.MBR; RUN;
PROC PRINT DATA=OUT.MBR (OBS=20); RUN;
  /* CHECK NO DUPLICATES */
PROC SORT DATA=OUT.MBR (KEEP=SSN) NODUPKEY OUT=TEMPS.CHECK;
  BY SSN;
RUN;
```

Appendix A.37
JCL: F814BAT

```
// $8043MRG JOB (12510000,T715,,SAS,,ITC9FL),8043LOEW,  
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043  
// *****  
// *  
// * *-----*  
// * | OPDR.TG.PRD.ETTW.N8043.LIB11(F814BAT) |  
// * *-----*  
// *  
// * 1. ALL STEPS ARE STEP RESTARTABLE  
// * 2. SAS PROGRAM CODE MUST BE SAVED IN A LIBRARY  
// * IN A MEMBER NAMED PROCMBR  
// *  
// * CONTACT MIRIAM LOEWENBERG  
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829  
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM  
// * PROCESS THE MBR FILES - JCL CALLS THE PROGRAM  
// *****  
// *  
// JS010 EXEC SAS9,  
// WORK='120000,60000'  
// *  
// IN DD DSN=OPDR.TG.PRD.ETTW.N8043.READ814.COMB10.SSD,DISP=SHR  
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.PROCMBR.D1012.SSD,  
// DISP=(OLD,CATLG,DELETE),  
// UNIT=TSILO,VOL=(,,10)  
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.N8043.LIB11(PROCMBR),  
// DISP=(SHR,PASS,KEEP)  
// *
```

Appendix A.38
SAS Code: PROCMBR

```

/*****
* FILENAME: PROCMBR
* PROGRAMMER:MIRIAM LOEWENBERG
* PURPOSE:To create MONTHLY VARS FROM MBR DATA FOR TTW
* add new variables to output file
* relabel dual eligible vars - forgot to give them new labels
* LAST EDITED: 11/29/11
*****/

*****/
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES ;

%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%LET ENDMN=12; /* CHANGE AS NEEDED */

/* STEP TO ASSIGN MACRO VARIABLES TO HANDLE TIME SERIES DATA */

%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
  %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
  %ELSE %IF &I < 2010 %THEN %LET YR=0%EVAL(&I-2000);
  %ELSE %LET YR=%EVAL(&I-2000);
  %IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
  %ELSE %LET EMN=12;
  %DO J=1 %TO &EMN;
    %IF &J<10 %THEN %LET MN=0%EVAL(&J);
    %ELSE %LET MN=%EVAL(&J);
    %LET X&K=%EVAL(&YR)%EVAL(&MN);
    %LET K=%EVAL(&K+1);
  %END;
%END;
%LET TOT=%EVAL(&K-1);

DATA OUT.MBR;
SET IN.MBR
  (RENAME=(DOEI=DOEIX DOEC=DOECX DOECD=DOECDX DOEID=DOEIDX))
;
BY SSN;

/* CREATE LONGER VERSIONS OF DOEI DOEC */
LENGTH DOEI DOEC DOECD DOEID 8;

%MACRO GETDATE (DT,NEWDT);
IF &DT GT .Z THEN DO;
YEAR = YEAR(&DT);
MONTH = MONTH(&DT);
DAY = DAY(&DT);

&NEWDT = MDY(MONTH,DAY,YEAR);
END;

%MEND;
```

Appendix A.38
SAS Code: PROCMBR

```

%GETDATE(DOECX,DOEC);
%GETDATE(DOEIX,DOEI);
%GETDATE(DOECDX,DOECD);
%GETDATE(DOEIDX,DOEID);

/* SET UP ARRAYS FOR CREATING VARIABLES */
/* CREATE ARRAYS FOR MONTHLY VARIABLES */
/* NEW VARIABLES */
ARRAY DX1{*} $10 %DO I=1 %TO &TOT; T2DX1&&X&I %END;;
ARRAY DX2{*} $10 %DO I=1 %TO &TOT; T2DX2&&X&I %END;;
ARRAY CDRN {*} $1 %DO I=1 %TO &TOT; CDR&&X&I %END;;
ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;

/* OLD VARIABLES */
ARRAY DIAG1{*} $ %DO I=1 %TO 12; DIG&I %END;;
ARRAY DIAG2{*} $ %DO I=1 %TO 12; SDIG&I %END;;
ARRAY ONS (*) %DO I = 1 %TO 12; DDO&I %END;; /* ONSET DATES */
ARRAY ENT (*) %DO I = 1 %TO 12; ENTDAT&I %END;;
/* DISABILITY ENTITLE DATE */
ARRAY CES (*) %DO I = 1 %TO 12; DDBC&I %END;;
/* DISABILITY CESS DATE */
ARRAY WRK (*) %DO I = 1 %TO 12; SDS&I %END;;
/* SGA CESS DATE */
ARRAY APS (*) %DO I = 1 %TO 12; APS&I %END;; /* APPEAL DATES */
ARRAY BDC (*) $ %DO I = 1 %TO 12; BDC&I %END;; /* DENIAL CODES */
ARRAY CDR (*) $ %DO I = 1 %TO 12; CDR&I %END;; /* DIS CESS REASON
*/

/* INITIALIZE MONTHLY VARS TO MISSING */
DO I = 1 TO &TOT;
    DX1(I) = ' ';
    DX2(I) = ' ';
    CDRN(I) = ' ';
END;

/* LOOP THROUGH THE NUMBER (NODF) OF DISABILITY FIELDS */
DO K = 1 TO NODF;

/* THE FOLLOWING ROUTINE CREATES MONTHLY REFERENCE DATES FROM
JAN 1994 TO DEC 2010 */
J=1;
Y=&BEGYR;

/* IF THERE IS A TERMINATION DATE IN LATEST ENTRY - END MONTHLIES
*/
DO I = 1 TO DIM(CYEAR); /* LOOP THROUGH ALL MONTH/YEAR REFERENCES
*/

CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
IF INT(I/12)=I/12 THEN DO;
    J=1;
    Y=Y+1;

```

Appendix A.38
SAS Code: PROCMBR

```
END;
ELSE J=J+1;

NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4)||SUBSTR(CYEAR(I),5,2),YMMN6.);

/* REVISION - IF THERE IS A DISABILITY CESSATION DATE THEN USE IT
TO LIMIT THE RANGE OF THE VARIABLES.
ELSE ALLOW THE RANGE TO EXTEND THE ENTIRE PERIOD OF DISABILITY
*/

IF ENT(K) > .Z THEN DO;
  IF ENT(K) LE NYEAR(I) AND CES(K) GT NYEAR(I) THEN DO;
    DX1(I) = DIAG1(K);
    DX2(I) = DIAG2(K);
    CDRN(I) = CDR(K);
  END;
  ELSE IF CES(K) = . AND ENT(K) LE NYEAR(I) THEN DO;
    DX1(I) = DIAG1(K);
    DX2(I) = DIAG2(K);
    CDRN(I) = CDR(K);
  END;

END; /* DATE FALLS WITHIN REFERENCE MONTH */
END; /* MONTH/YEAR REFERENCE VARIABLES */
END; /* DISABILITY FIELDS (DO K= 1 TO NODF) */

LABEL
%DO I=1 %TO &TOT;
  T2DX1&&X&I = "&&X&I T2 PRIMARY DIAGNOSIS"
  T2DX2&&X&I = "&&X&I T2 SECONDARY DIAGNOSIS"
  CDR&&X&I = "&&X&I T2 CESSATION OF DISABILITY REASON"
%END;
%DO I = 1 %TO &TOT; /* START OCTOBER 2004 */
  WIC&&X&I = "&&X&I T2 WORK INDICATION CODE"
  RFD&&X&I = "&&X&I T2 REASON FOR DEDUCTION"
  BPD&&X&I = "&&X&I T2 BENEFIT PAYMENT DESIGNATION"
  RFST&&X&I = "&&X&I T2 REASON FOR SUSPENSION/TERMINATION"
%END;
%DO I=1 %TO 12;
  DIG&I = "T2 PRIMARY DIAGNOSIS ENTRY &I"
  SDIG&I = "T2 SECONDARY DIAGNOSIS ENTRY &I"
  DDO&I = "T2 DISABILITY ONSET ENTRY &I"
  ENTDAT&I = "T2 DISABILITY ENTITLEMENT DATE ENTRY &I"
  DDBC&I = "T2 DISABILITY CESSATION DATE ENTRY &I"
  APS&I = "T2 APPEALS DATE ENTRY &I"
  BDC&I = "T2 DENIAL CODES ENTRY &I"
  HDD&I = "T2 HEARING DECISION DATE &I"
  CSA&I = "T2 CURRENT SGA ACTIVITY ENTRY &I"
  CDR&I = "T2 DISABILITY CESSATION REASON ENTRY &I"
  DSD&I = "T2 DISABILITY ADJUDICATION DATE ENTRY &I"
  DAC&I = "T2 DISABILITY AWARD CODE ENTRY &I"
```

Appendix A.38
SAS Code: PROCMBR

```
EBD&I = "T2 EPE BEGIN DATE ENTRY &I"  
LOD&I = "T2 LEVEL OF DENIAL CODE ENTRY &I"  
SDS&I = "T2 SGA DISABILITY CESSATION ENTRY &I"  
%END;  
;  
DROP I J K Y  
%DO I = 1 %TO &TOT; C&&X&I N&&X&I %END;;  
DROP DOECX DOEIX DOECDX DOEIDX YEAR MONTH DAY;  
  
LABEL DOEC = 'DATE OF CURRENT ENTITLEMENT-SSDI'  
DOEI = 'DATE OF INITIAL ENTITLEMENT-SSDI'  
DOECD = 'DATE OF CURRENT ELIG-DUALELIG-SUPPLEMENTARY RECORD'  
DOEID = 'DATE OF INITIAL ELIG-DUALELIG-SUPPLEMENTARY RECORD'  
DDOD = 'DUAL ELIG-DATE OF DISABILITY ONSET-SUPPLEMENTARY RECORD'  
DACD = 'DUAL ELIG-DISABILITY AWARD CODE-SUPPLEMENTARY RECORD'  
LODD = 'DUAL ELIG-LEVEL OF DENIAL CODE-SUPPLEMENTARY RECORD'  
ENTDATD = 'DUAL ELIG-DATE OF ENTITLEMENT-SUPPLEMENTARY RECORD'  
DDBCDD = 'DUAL ELIG-DATE OF BENEFIT CESSATION-SUPPLEMENTARY RECORD'  
DSDD = 'DUAL ELIG-ADJUDICATION DATE-SUPPLEMENTARY RECORD'  
HDDD = 'DUAL ELIG-HEARING DECISION DATE-SUPPLEMENTARY RECORD'  
SDSD = 'DUAL ELIG-SGA DISABILITY CESSATION DATE-SUPPLEMENTARY  
RECORD'  
ADCD = 'DUAL ELIG-APPLICANT DISAB CESSATION DATE-SUPPLEMENTARY  
RECORD'  
APSD = 'DUAL ELIG-APPEALS DATE-SUPPLEMENTARY RECORD'  
EBDD = 'DUAL ELIG-EPE BEGIN DATE-SUPPLEMENTARY RECORD'  
BDCD = 'DUAL ELIG-BASIS FOR DENIAL CODE-SUPPLEMENTARY RECORD'  
CDRD = 'DUAL ELIG-DISABILITY CESSATION REASON-SUPPLEMENTARY RECORD'  
CSAD = 'DUAL ELIG-CURRENT SGA STATUS-SUPPLEMENTARY RECORD'  
DIGD = 'DUAL ELIG-PRIMARY DIAGNOSIS-SUPPLEMENTARY RECORD'  
SDIGD = 'DUAL ELIG-SECONDARY DIAGNOSIS-SUPPLEMENTARY RECORD'  
;  
RUN;  
PROC PRINT DATA=OUT.MBR(OBS=25);  
FORMAT ENTDAT1-ENTDAT12  
DDBC1-DDBC12 DSD1-DSD12 HDD1-HDD12 SDS1-SDS12  
ADC1-ADC12 APS1-APS12 DDO1-DDO12 EBD1-EBD12  
DOEI DOEC  
DOST PIED1-PIED50  
SMI_STAR SMI_TERM YYMMN6.;  
TITLE 'MBR DATA'; RUN;  
PROC CONTENTS; RUN;  
%MEND;  
%START;
```

Appendix A.39
JCL/SAS Code: MRGPBEN

```
// $8043MRG JOB (11710000,T715,,SAS,,ITC9FL),8043MIR,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043
// *****
// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.N8043.LIB11(MRGPBEN)      |
// *          *-----*
// * STEP1 -MERGE SUMRECS WITH DPEN AND PBEN ONTO MBR RECORDS
// * INPUT FILES ARE SUMMED RECORDS FROM MBRADDUP AND
// * ALL MBR RECORDS OUTPUT FROM PROCMBR
// * CAN DO IT BY SSN BECAUSE SUMRECS IS ONLY PRIMARIES
// * THE CAN = SSN FOR THESE CASES
// * CONTACT MIRIAM LOEWENBERG
// * SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
// * E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
// *****
// *
// * SET REG='128M'
// JS010 EXEC SAS9,
//          WORK='200000,100000'          * SPACE IN BLKS *
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPHUS.SUMREC.Y2010,DISP=SHR
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.PROCMBR.D1012.SSD,DISP=SHR
// OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
// SYSIN DD *
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;
DATA OUT.MBR;
MERGE IN2.MBR (IN=M1) IN1.MBR
;
BY SSN ;
IF M1;
RUN;

PROC CONTENTS; RUN;
PROC PRINT DATA=OUT.MBR (OBS=25); RUN;
```

Appendix A.40
JCL: JCLLONG1

```
//$4671L1 JOB (12510000,T715,,SAS,,ITC9FL),4671MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*          RESTART=JS020.SAS9
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(JCLLONG) |
//*          *-----*
//*
//*          1. READ IN EACH SEGMENT OF SSI LONGITUDINAL FILE
//*          2. SORT BY SSN AND RCD_EST AND OUTPUT STEP 1
//* JCL CALLS THE PROGRAM FOR STEP 1 RLONGST1
//* CONTACT DAWN PHELPS
//* SSA PHONE 202 358-6316 MPR PHONE 202 554-7556
//* E-MAIL DPHELPS@MATHEMATICA-MPR.COM
//*****
//*
//JS010 EXEC SAS9,
//          WORK='120000,60000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P1D1012,
//          DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1 DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2 DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P1,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//          WORK='120000,60000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P2D1012,
//          DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1 DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2 DD DSN=*&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P2,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
//          WORK='120000,60000'          * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P3D1012,
```

Appendix A.40
JCL: JCLLONG1

```
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P3,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS040    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P4D1012,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P4,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS050    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P5D1012,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P5,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS060    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P6D1012,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
```

Appendix A.40
JCL: JCLLONG1

```
//TEMP2 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P6,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
// DISP=(SHR,PASS,KEEP)
//*
//JS070 EXEC SAS9,
// WORK='120000,60000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P7D1012,
// DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
// DISP=SHR
//TEMP DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P7,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
// DISP=(SHR,PASS,KEEP)
//*
//JS080 EXEC SAS9,
// WORK='120000,60000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P8D1012,
// DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
// DISP=SHR
//TEMP DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P8,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
// DISP=(SHR,PASS,KEEP)
//*
//JS090 EXEC SAS9,
// WORK='120000,60000' * SPACE IN BLKS *
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P9D1012,
// DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
// DISP=SHR
//TEMP DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P9,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
```

Appendix A.40
JCL: JCLLONG1

```
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS100    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '          * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P10D1012 ,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT ,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP1    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP2    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P10 ,
//          DISP=( OLD ,CATLG ,DELETE ) ,
//          SPACE=( CYL , ( 1000 ,100 ) ,RLSE ) ,VOL=( , , ,10 )
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1) ,
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS110    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '          * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P11D1012 ,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT ,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP1    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP2    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P11 ,
//          DISP=( OLD ,CATLG ,DELETE ) ,
//          SPACE=( CYL , ( 1000 ,100 ) ,RLSE ) ,VOL=( , , ,10 )
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1) ,
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS120    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '          * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P12D1012 ,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT ,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP1    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//TEMP2    DD DSN=&&TEMP ,DISP=( NEW ,DELETE ,DELETE ) ,UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P12 ,
//          DISP=( OLD ,CATLG ,DELETE ) ,
//          SPACE=( CYL , ( 1000 ,100 ) ,RLSE ) ,VOL=( , , ,10 )
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1) ,
//          DISP=( SHR ,PASS ,KEEP )
//*
//JS130    EXEC SAS9 ,
//          WORK=' 120000 ,60000 '          * SPACE IN BLKS *
//*
```

Appendix A.40
JCL: JCLLONG1

```
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P13D1012,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P13,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
//JS140    EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.LONG2010.P14D1012,
//          DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,
//          DISP=SHR
//TEMP     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P14,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,,10)
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1),
//          DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.41
SAS Code: RLONGST1

```
/* *****  
 * FILENAME: OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST1)  
 * PROGRAMMER: DAWN PHELPS (FROM MIRIAM LOEWENBERG)  
 * PURPOSE: TO READ IN THE LONGITUDINAL FILE  
 * CREATED: 4/17/07  
*****/  
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;  
%LET BEGYR=1994;  
%LET ENDYR=2010;  
%LET ENDMN=12;  
  
/* NOTE: CHANGE THE ENDYR AND ENDMN PARAMETER BEFORE EACH RUN */  
  
/* ASSIGN MACROS VARIABLES TO HANDLE TIME SERIES DATA */  
  
%MACRO START;  
  %LET K=1;  
  %DO I=&BEGYR %TO &ENDYR;  
    %IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);  
    %ELSE %IF %EVAL(&I-2000)<10 %THEN %LET YR=0%EVAL(&I-2000);  
    %ELSE %LET YR=%EVAL(&I-2000);  
    %IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);  
    %ELSE %LET EMN=12;  
    %DO J=1 %TO &EMN;  
      %IF &J<10 %THEN %LET MN=0%EVAL(&J);  
      %ELSE %LET MN=%EVAL(&J);  
      %LET X&K=%EVAL(&YR)%EVAL(&MN);  
      %LET K=%EVAL(&K+1);  
    %END;  
  %END;  
%LET TOT=%EVAL(&K-1);  
  
/* *****  
  READ IN EVERY RECORD IN THIS SECTION OF THE FILE  
  AFTER ALL SECTIONS ARE READ IN THEN PROCESS THE MULTIPLE RECORDS  
  SORT EACH SECTION BY SSN AND FILE SELECTION DATE TO ENSURE  
  PROPER GROUPING OF MULTIPLE RECORDS IN THE FINAL COMBINE STEP  
HE  
*****  
***/  
/* READ IN THE FINDER RECORDS TO USE AS A FILTER */  
/* READ IN THE FINDER RECORDS TO USE AS A FILTER */  
DATA FINDER;  
  INFILE IN2 TRUNCOVER;  
  INPUT @001 SSN $9.  
  ;  
RUN;  
PROC SORT; BY SSN; RUN;  
  
DATA TEMP.LONG;
```

Appendix A.41
SAS Code: RLONGST1

```

INFILE IN1 TRUNCOVER;

INPUT
@01 HUN          $9.          /* HOUSED UNDER NUMBER */
@10 SSN          $9.          /* PAN SOCIAL SECURITY NUMBER
*/
@19 TOA          $2.          /* TOA TYPE OF ACTION
*/
@21 COMP_STA    $2.          /* COMP-STAT-TOA COMP. STATUS TYPE OF
ACTION */
@23 MFT          $2.          /* MFT MASTER FILE TYPE CODE
*/
@25 START_RD    ?? YMMN6.    /* START-RD EARLIEST COMPUTATION DATE
*/
@41 RCD_EST     ?? YMMDD8.    /* RCD-EST-JD RECORD ESTABLISHMENT DATE
*/
@49 BIRTH_JD    ?? YMMDD8.    /* BIRTH-JD DATE OF BIRTH
*/
@57 DEATH_JD    ?? YMMDD8.    /* DEATH-JD DATE OF DEATH
*/
@65 LAF         $2.          /* LAF MBR LEDGER ACCOUNT FILE CODE
*/
@67 CURSTAT     $3.          /* CURSTAT CURRENT PAYMENT STATUS
*/
@70 SEX         $1.          /* SEX SEX
*/
@71 RACE        $1.          /* RACE RACE
*/
@72 ELG_RD      ?? YMMN6.    /* ELG-RD DATE OF CURRENT ELIGIBILITY
*/
@78 APP_RD      ?? YMMDD8.    /* APPL-JD APPLICATION DATE (CURRENT)
*/
@86 _8080_JD    ?? YMMDD8.    /* 8080-JD 8080 DATE
*/
@112 CLM_FIL    ?? YMMDD8.    /* CLM-FIL-JD DATE CLAIM FILED
*/
@120 DISPAYCD   $1.          /* DISABILITY PAYMENT CODE
*/
@136 DIB_DIG    $4.          /* DIB-DIG PRIMARY DISABILITY DIAGNOSTIC
CODE */
@140 DIB_DIG2   $4.          /* DIB-DIG2 SECONDARY DISABILITY
DIAGNOSTIC CODE*/
@144 DIB_MDR    $1.          /* DIB-MDR MEDICAL DIARY REASON
*/
@145 DIB_DPM    $1.          /* PERMANENT DISABILITY INDICATOR
*/
@146 PDSCC      $6.          /* PDSCC RESIDENCE STATE/COUNTY/DO CODE
*/
@152 PDZIP      $5.          /* PDZIP PAYEE'S ZIP CODE
*/
@157 PDZIP6_9   $4.          /* PDZIP6-9 PAYEE'S ZIP CODE SUFFIX
*/

```

Appendix A.41
SAS Code: RLONGST1

```

@161 DO                $3.                /* DO DISTRICT OFFICE CODE
*/
@172 REPPAYTP          $3.                /* REPPAYTYP TYPE OF PAYEE CODE
*/
@197 FIRST_PA          ?? YMMDD8.         /* FIRST-PAY-DTE FIRST PAYMENT DATE
(THIS RECORD)*/
@217 STOP_RD           ??  YMMN6.         /* STOP-RD STOP DATE
*/
@223 LANG_WR           $2.                /* LANGUAGE PREFERENCE WRITTEN
*/
@;

POS=10097;
%DO I=1 %TO &TOT;
  INPUT
  @POS+6  PST&&X&I  $3.  /* PSTAT PAYMENT STATUS CODE          */
  @POS+9  LVF&&X&I  $1.  /* LIVF LIVING ARRANGEMENT CODE */
  @POS+10 STC&&X&I  $1.  /* STCONCATM STATE CONCURR ELIG IND*/
  @POS+11 TKT&&X&I  $1.  /* TICKET STATUS INDICATOR          */
  @POS+12 EIN&&X&I   4.  /* EINCM CHARGEABLE EARNED INCOME AMOUNT
*/
  @POS+16 UIN&&X&I   4.  /* UINCM CHARGEABLE UNEARNED INCOME
AMOUNT */
  @POS+20 FDA&&X&I   3.  /* FEDAMT FEDERAL ASSISTANCE AMOUNT */
  @POS+23 SPA&&X&I   3.  /* SUPAMT CURRENT AMOUNT OF STATE
SUPPLEMENT */
  @POS+26 MDT&&X&I   $1. /* MEDTEST MEDICAL AND SOCIAL SERVICE
INCOME TEST*/
  @POS+28 FDP&&X&I   6.  /* FEDPMT FEDERAL MONEY AMOUNT (PAYMENT)
*/
  @POS+34 STP&&X&I   6.  /* STATPMT STATE SUPPL AMOUNT
(PAYMENT)*/
  @;
  POS=POS+41;
%END;

/* MARY BARBOUR INSTRUCTION */
IF SUBSTR(MFT,1,1) = 'X' THEN DELETE;

DROP POS;

LENGTH
%DO I=1 %TO &TOT;
  EIN&&X&I
  UIN&&X&I
  FDA&&X&I
  SPA&&X&I
  STP&&X&I
  FDP&&X&I
%END;

_8080_JD APP_RD BIRTH_JD CLM_FIL DEATH_JD ELG_RD FIRST_PA RCD_EST
START_RD STOP_RD

```

Appendix A.41
SAS Code: RLONGST1

4;

```
FORMAT  
START_RD STOP_RD MONYY7.  
RCD_EST BIRTH_JD DEATH_JD _8080_JD CLM_FIL FIRST_PA APP_RD ELG_RD  
MMDDYY10.
```

```
;  
RUN;
```

```
PROC PRINT DATA=TEMP.LONG (OBS=25); TITLE 'BEFORE SCREEN';RUN;  
PROC CONTENTS; RUN;  
PROC SORT DATA=TEMP.LONG OUT=TEMP1.LONG; BY SSN; RUN;
```

```
DATA TEMP2.LONG;  
MERGE TEMP1.LONG (IN=T) FINDER (IN=F);  
BY SSN;  
IF T AND F;  
RUN;  
/* SORT EACH SEGMENT BY SSN AND RECORD ESTAB DATE FOR COMBINING */
```

```
PROC SORT DATA=TEMP2.LONG OUT=OUT1.LONG; BY SSN RCD_EST; RUN;  
PROC PRINT DATA=OUT1.LONG (OBS=25); TITLE 'AFTER SCREEN';RUN;  
%MEND;  
%START;
```

Appendix A.42
JCL/SAS Code: COMBLONG

```

//$4671LC JOB (12510000,T715,,SAS,,ITC9FL),4671MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(COMBLONG) |
//*          *-----*
//*
//*          1.  COMBINE EACH SEGMENT OF SSI LONGITUDINAL FILE
//*          2.  SORTED BY SSN AND RCD_EST AND OUTPUT COMBINED FILE
//*          3.  DE-DUP BY RECORD ESTABLISHMENT DATE
//*****
//*
//JS030     EXEC SAS9,
//          WORK='200000,100000'                * SPACE IN BLKS *
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P1,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P2,DISP=SHR
//IN3      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P3,DISP=SHR
//IN4      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P4,DISP=SHR
//IN5      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P5,DISP=SHR
//IN6      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P6,DISP=SHR
//IN7      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P7,DISP=SHR
//IN8      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P8,DISP=SHR
//IN9      DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P9,DISP=SHR
//IN10     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P10,DISP=SHR
//IN11     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P11,DISP=SHR
//IN12     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P12,DISP=SHR
//IN13     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P13,DISP=SHR
//IN14     DD DSN=OPDR.TG.PRD.ETTW.N4671.LONG2010.P14,DISP=SHR
//*
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.N4671.COMBLONG.D1012.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//*
//SYSIN    DD *

OPTIONS NOCENTER COMPRESS=YES;
DATA OUT1.LONG;
  SET IN1.LONG
      IN2.LONG
      IN3.LONG
      IN4.LONG
      IN5.LONG
      IN6.LONG
      IN7.LONG
      IN8.LONG
      IN9.LONG
      IN10.LONG
      IN11.LONG
      IN12.LONG
      IN13.LONG
      IN14.LONG;

```

Appendix A.42
JCL/SAS Code: COMBLONG

```
BY SSN RCD_EST;

/* IF A BENEFICIARY HAS MORE THAN 1 PERIOD OF ELIGIBILITY THERE
WILL
BE MULTIPLE RECORDS FOR HIM. THESE RECORDS MAY ACTUALLY APPEAR
IN
DIFFERENT SEGMENTS COMBINED ABOVE. THEREFORE ALL THE RECORDS
HAVE
TO BE ARRANGED ACCORDING TO THE RECORD ESTABLISHMENT DATE SO
THAT THEY CAN BE PROCESSED TO CREATE 1 RECORD FOR EACH
BENEFICIARY
THERE ARE SOME DUPLICATE RECORDS ON THE FILE WHICH ARE DELETED
THERE ARE DUPLICATE RECORDS IN THE FILE SINCE A HOUSEHOLD MEMBER
MAY APPEAR ASSOCIATED WITH ANOTHER HOUSEHOLD MEMBER IN DIFFERENT
SEGMENTS OF THE FINDER MODULES.
THE FILE MUST BE DE-DUPLICATED BY RECORD ESTABLISHMENT DATE */

/*DEDUP */
IF LAST.RCD_EST;

RUN;
PROC PRINT DATA=OUT1.LONG (OBS=25); RUN;
PROC CONTENTS; RUN;
```

Appendix A.43

JCL: JCLLNGS2

```

//$4671L2 JOB (12510000,T715,,SAS,,ITC9FL),4671MJL,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*
//*          *-----*
//*          |  OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(JCLLNGS2)  |
//*          *-----*
//*
//*          1.  PROCESS COMBINED  LONGITUDINAL FILE
//*          2.  OUTPUT UPDATED STEP2 FILE
//*
//* SEGMENTS READ IN IN STEP 1 WERE COMBINED USING PROGRAM COMBLONG
//* THE COMBINED RECORDS WERE DE-DUPED BY SSN RCD_EST
//* JCL CALLS PROGRAM RLONGST2
//*****
//*
//JS030     EXEC SAS9,
//          WORK='120000,60000'                * SPACE IN BLKS *
//*
//IN1       DD DSN=OPDR.TG.PRD.ETTW.N4671.COMBLONG.D1012.SSD,DISP=SHR
//TEMP      DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TEMP1     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TVF       DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TVG       DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TVB1      DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TVB2      DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(3000,3000),RLSE),VOL=(,,20)
//TEMP3     DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,20)
//OUT1      DD DSN=OPDR.TG.PRD.ETTW.N4671.RLONGST2.D1012.SSD,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//*
//SYSIN     DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.SSR.PRDLIB(RLONGST2),
//          DISP=(SHR,PASS,KEEP)
//*

```

Appendix A.44

SAS Code: RLONGST2

```

/*****
* FILENAME: RLONGST2.SAS
* PROGRAMMER: MIRIAM LOEWENBERG
* PURPOSE: TO PROCESS THE LONGITUDINAL FILE
* CREATED: 2/14/05
* REVISED RACHEL SULLIVAN - NAZMUL KHAN'S SPI PROGRAMS
* REVISED DEDUPLICATED FILE APRIL 2009
*****/

*****/
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY;
%LET BEGYR=1994;
%LET ENDYR=2010;
%LET ENDMN=12;

/* NOTE: CHANGE THE ENDYR AND ENDMN PARAMETER BEFORE EACH RUN */

/* ASSIGN MACROS VARIABLES TO HANDLE TIME SERIES DATA */

%MACRO START;
%LET K=1;
%DO I=&BEGYR %TO &ENDYR;
%IF &I<2000 %THEN %LET YR=%EVAL(&I-1900);
%ELSE %IF %EVAL(&I-2000)<10 %THEN %LET YR=0%EVAL(&I-2000);
%ELSE %LET YR=%EVAL(&I-2000);
%IF &I=&ENDYR %THEN %LET EMN=%EVAL(&ENDMN);
%ELSE %LET EMN=12;
%DO J=1 %TO &EMN;
%IF &J<10 %THEN %LET MN=0%EVAL(&J);
%ELSE %LET MN=%EVAL(&J);
%LET X&K=%EVAL(&YR)%EVAL(&MN);
%LET K=%EVAL(&K+1);
%END;
%END;
%LET TOT=%EVAL(&K-1);

/*****
WE NEED TO READ THE SSI LONG FILES TO CAPTURE THE HISTORY FIELDS
WE NEED TO SCAN ALL THE RECORDS FOR VALID VALUES INSTEAD OF JUST
KEEPING THE LAST RECORD AS WE DID PREVIOUSLY.
THE LAST RECORD MAY HAVE HISTORY FIELDS BLANKED OUT
*****/
***/

/*SPLIT THE VARIABLE FILE TO UPDATE RECORDS WITH NONMISSING DATA
*/

DATA TEMP.LONGF(KEEP=HUN--LANG_WR) /* FIXED PART */
TEMP1.LONGV(KEEP=SSN RCD_EST /* VARIABLE PART */
%DO I=1 %TO &TOT;
PST&&X&I LVF&&X&I STC&&X&I EIN&&X&I UIN&&X&I FDA&&X&I
SPA&&X&I
MDT&&X&I FDP&&X&I STP&&X&I TKT&&X&I

```

Appendix A.44
SAS Code: RLONGST2

```
%END;);
SET IN1.LONG; BY SSN RCD_EST;
IF SSN = '000000000' THEN DELETE;
RUN;

PROC PRINT DATA=TEMP.LONGF (OBS=20); TITLE 'BEFORE UPDATE'; RUN;
PROC PRINT DATA=TEMP1.LONGV (OBS=50); RUN;
PROC FREQ DATA=TEMP.LONGF; TABLES SEX RACE CURSTAT; RUN;
PROC FREQ DATA=TEMP1.LONGV; TABLES PST0901 PST0912 LVF0901 LVF0912;
RUN;

/*CREATE MIN AND MAX VALUES FROM SOME OF THE FIXED PART VARIABLES
*/

PROC MEANS NOPRINT DATA=TEMP.LONGF;
VAR START_RD RCD_EST ELG_RD APP_RD _8080_JD CLM_FIL FIRST_PA
STOP_RD;
BY SSN;
OUTPUT OUT=TEMP3.LONG2(DROP=_)
MIN=MINSTRRD MINRCDST MINELGRD MINAPPRD MIN_80JD MINCLMFL
MINFRSTP MINSTPRD
MAX=MAXSTRRD MAXRCDST MAXELGRD MAXAPPRD MAX_80JD MAXCLMFL
MAXFRSTP MAXSTPRD
;
RUN;

/*SPLIT THE FIXED PART FILE TO UPDATE RECS WITH NONMISSING DATA */
/*FOR FIXED DATA WE DO NOT HAVE TO EXAMINE INTERMEDIATE RECORDS */
/*WE WANT THE CURRENT VALUE IN THE FIELDS */

DATA TVG.LONGFA TVF.LONGFB;
SET TEMP.LONGF;
BY SSN RCD_EST;
IF FIRST.SSN THEN OUTPUT TVG.LONGFA;
ELSE OUTPUT TVF.LONGFB;
RUN;

PROC DATASETS LIBRARY=TEMP;
DELETE LONGF;
RUN;
/* RUN THE UPDATE PROCEDURE */

DATA TEMP.LONGF;
UPDATE TVG.LONGFA TVF.LONGFB;
BY SSN;
RUN;

/* CONVERSATIONS WITH MARY BARBOUR HAVE REVEALED THAT SSA WILL
TRY TO SAVE SPACE BY BLANKING AND ZEROING OUT HISTORY FIELDS
WHEN A NEW PERIOD OF ELIGIBILITY IS ESTABLISHED FOLLOWING THE
END OF AN OLD PERIOD OF ELIGIBILITY (CREATING A NEW RECORD).
THEREFORE WE CAN NOT TAKE THE HISTORY VALUES FROM THE LAST RECORD
AS PREVIOUSLY BUT MUST SCAN ALL THE RECORDS USING AN UPDATE
```

Appendix A.44

SAS Code: RLONGST2

```
PROCEDURE.  FOR CHARACTER VALUES, A BLANK ON A SUBSEQUENT RECORD
(WHERE THERE HAD BEEN A VALUE ON A PREVIOUS RECORD) WILL NOT
OVERWRITE THE VALUE USING THE UPDATE PROCEDURE.  HOWEVER, FOR
NUMERIC VALUES, SSA WILL ZERO OUT THE FIELD AND UPDATE WILL
OVERWRITE A VALUE WITH THE ZERO.  AS A RESULT WE WILL BLANK OUT
THESE ZEROS SO AS NOT TO OVERWRITE THE FIELD.
WITH REFERENCE TO THE ACTUAL STATE AND FEDERAL PAYMENT FIELDS
MARY HAS ALSO INDICATED THAT THERE MAY BE DOLLAR ADJUSTMENTS TO A
GIVEN MONTHLY PAYMENT ON A SUBSEQUENT RECORD.  FOR EXAMPLE,
FDP0309 (FED PAYMENT IN SEPT 2003) MAY SHOW PAYMENTS ON 2 RECORDS
THE TOTAL ACTUAL PAYMENT BEING THE SUM OF THESE.  THIS IS HANDLED
IN THE FOLLOWING PROCEDURE  */
PROC DATASETS LIBRARY=TVG;
  DELETE LONGFA;
RUN;

PROC DATASETS LIBRARY=TVF;
  DELETE LONGFB;
RUN;

DATA TVG.LONGVA TVF.LONGVB;
  SET TEMP1.LONGV; BY SSN RCD_EST;

  /* ADD UP THE PAYMENT AMOUNTS ACROSS RECORDS */
  /* IN THE CASE WHERE A FIELD IS ADJUSTED, THE INCREMENT
  WILL APPEAR ACROSS RECORDS AND THE UPDATE PROCEDURE
  WILL TAKE THE SUM INSTEAD OF OVERWRITING ONE AMOUNT
  WITH A SUBSEQUENT AMOUNT */

  ARRAY FPT (*) %DO I=1 %TO &TOT; FDP&&X&I %END;;
  ARRAY SPT (*) %DO I=1 %TO &TOT; STP&&X&I %END;;
  ARRAY FMP (*) %DO I=1 %TO &TOT; FPMT&&X&I %END;;
  ARRAY SDP (*) %DO I=1 %TO &TOT; SPMT&&X&I %END;;

  RETAIN %DO I=1 %TO &TOT; FPMT&&X&I %END;;
  RETAIN %DO I=1 %TO &TOT; SPMT&&X&I %END;;

  DO I = 1 TO DIM(FPT);
    IF FIRST.SSN THEN DO;
      FMP(I) = FPT(I);
      SDP(I) = SPT(I);
    END;
    ELSE DO;
      IF FPT(I) >= 0 THEN FMP(I) + FPT(I);
      IF SPT(I) >= 0 THEN SDP(I) + SPT(I);
    END;
  END;
  DROP %DO I=1 %TO &TOT; FDP&&X&I %END;;
  DROP %DO I=1 %TO &TOT; STP&&X&I %END;;

  /* RENAME PAYMENT VARIABLES BACK TO ORIGINAL NAME */

  %DO I = 1 %TO &TOT;
```

Appendix A.44
SAS Code: RLONGST2

```
        RENAME FPMT&&X&I = FDP&&X&I;
        RENAME SPMT&&X&I = STP&&X&I;
%END;

IF FIRST.SSN THEN OUTPUT TVG.LONGVA;
ELSE OUTPUT TVF.LONGVB;
RUN;
/*
PROC SORT DATA=TVB.LONGVB OUT=TVB1.LONGVB; BY SSN RCD_EST; RUN;
*/
DATA TVB2.LONGVB;
  SET TVF.LONGVB; BY SSN RCD_EST;

/* ON THE LAST RECORD IF IT IS A TERMINATION RECORD THE WHOLE
   HISTORY MAY BE BLANKED AND ZEROED */

ARRAY FDA (*) %DO I=1 %TO &TOT; FDA&&X&I %END;;
ARRAY SPA (*) %DO I=1 %TO &TOT; SPA&&X&I %END;;
ARRAY EIC (*) %DO I=1 %TO &TOT; EIN&&X&I %END;;
ARRAY UIC (*) %DO I=1 %TO &TOT; UIN&&X&I %END;;
ARRAY FPT (*) %DO I=1 %TO &TOT; FDP&&X&I %END;;
ARRAY SPT (*) %DO I=1 %TO &TOT; STP&&X&I %END;;

/* FLAG THE NUMERIC FIELDS IF GT 0 ON ANY RECORD -
   NEVER SET THEM TO ZERO IF THEY ARE ZEROED OUT SUBSEQUENTLY */

ARRAY FAF (*) 3 %DO I=1 %TO &TOT; FAF&&X&I %END;;
ARRAY SAF (*) 3 %DO I=1 %TO &TOT; SAF&&X&I %END;;
ARRAY EIF (*) 3 %DO I=1 %TO &TOT; EIF&&X&I %END;;
ARRAY UIF (*) 3 %DO I=1 %TO &TOT; UIF&&X&I %END;;
ARRAY FPF (*) 3 %DO I=1 %TO &TOT; FPF&&X&I %END;;
ARRAY SPF (*) 3 %DO I=1 %TO &TOT; SPF&&X&I %END;;

RETAIN %DO I=1 %TO &TOT; FAF&&X&I
        SAF&&X&I
        EIF&&X&I
        UIF&&X&I
        FPF&&X&I
        SPF&&X&I %END;;

/* INITIALIZE FLAGS TO ZERO */
IF FIRST.SSN THEN DO;
  DO I = 1 TO DIM(FAF);
    FAF(I) = 0;
    SAF(I) = 0;
    EIF(I) = 0;
    UIF(I) = 0;
    FPF(I) = 0;
    SPF(I) = 0;
  END;
END;

DO I = 1 TO DIM(FDA);
```

Appendix A.44
SAS Code: RLONGST2

```
IF FDA(I) > 0 THEN FAF(I) = 1;
IF SPA(I) > 0 THEN SAF(I) = 1;
IF EIC(I) > 0 THEN EIF(I) = 1;
IF UIC(I) > 0 THEN UIF(I) = 1;
IF FPT(I) > 0 THEN FPF(I) = 1;
IF SPT(I) > 0 THEN SPF(I) = 1;
END;

/* IF THE AMOUNTS HAVE BEEN ZEROED OUT ON SUBSEQUENT RECORDS
WITH A NEW START DATE -
SET THE ZERO TO MISSING IF THE MONTHLY AMOUNT HAS BEEN
FLAGGED AS EVER BEING > 0. THIS PREVENTS THE UPDATE
PROCEDURE FROM OVERWRITING A RECORD WITH ZERO.
FOR THE ALPHANUMERIC FIELDS A BLANK WILL NOT OVERWRITE
A PREVIOUS VALID VALUE */

DO I = 1 TO DIM(FDA);
IF FDA(I) = 0 AND FAF(I) = 1 THEN FDA(I) = .;
IF SPA(I) = 0 AND SAF(I) = 1 THEN SPA(I) = .;
IF EIC(I) = 0 AND EIF(I) = 1 THEN EIC(I) = .;
IF UIC(I) = 0 AND UIF(I) = 1 THEN UIC(I) = .;
IF FPT(I) = 0 AND FPF(I) = 1 THEN FPT(I) = .;
IF SPT(I) = 0 AND SPF(I) = 1 THEN SPT(I) = .;
END;
RUN;

PROC DATASETS LIBRARY=TVF;
DELETE LONGVB;
RUN;
/* RUN THE UPDATE PROCEDURE */

DATA TVF.LONGV;
UPDATE TVG.LONGVA TVB2.LONGVB;
BY SSN;
DROP %DO I=1 %TO &TOT;FAF&&X&I
      SAF&&X&I
      EIF&&X&I
      UIF&&X&I
      FPF&&X&I
      SPF&&X&I %END; I;
RUN;

DATA OUT1.LONG;
MERGE TEMP.LONGF(IN=A)
      TEMP3.LONG2(IN=B)
      TVF.LONGV(IN=C)
      ;
BY SSN;
IF A;

RUN;

PROC PRINT DATA=OUT1.LONG (OBS=20); TITLE 'AFTER UPDATE'; RUN;
```

Appendix A.44
SAS Code: RLONGST2

```
PROC FREQ DATA=OUT1.LONG; TABLES SEX RACE CURSTAT  
PST1001 PST1012 LVF1009 LVF1012;  
RUN;  
PROC CONTENTS;  
RUN;  
  
%MEND;  
%START;
```

Appendix A.45
JCL/SAS Code: T16MRG31

```
//$8043T16 JOB (12510000,T715,,SAS,,ITC9FL),8043LOEW,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$8043  
//*****  
//*  
//*      *-----*  
//*      | AIS.P1171.$8043.LIB11(T16MRG31)      |  
//*      *-----*  
//*  
//*      1. ALL STEPS ARE STEP RESTARTABLE  
//*  
//* CONTACT MIRIAM LOEWENBERG  
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829  
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM  
//* MERGE 831-33 SSI DATA TO THE SSI LONGITUDINAL FILE  
//*****  
//*  
//JS010 EXEC SAS9,  
//      WORK='120000,60000'  
//*  
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N4671.RLONGST2.D1012.SSD,DISP=SHR  
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.SSICDR.SSD,DISP=SHR  
//OUT DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,  
//      DISP=(NEW,CATLG,DELETE),  
//      UNIT=TSILO,VOL=(,,10)  
//SYSIN DD *  
OPTIONS NOCENTER COMPRESS=BINARY;  
DATA OUT.LONG;  
MERGE IN1.LONG (IN=L) IN2.SSICDR ; BY SSN;  
IF L;  
RUN;  
PROC CONTENTS DATA=OUT.LONG; RUN;  
PROC PRINT DATA=OUT.LONG (OBS=25); RUN;
```

Appendix A.46
JCL: NUMJCL

```
//#2127NUM JOB (12510000,T715,,SAS,,ITC9FL),2127JLP,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//* !!! ONLY TO START A LATER JOB /* ,RESTART=JS230.SAS9 */
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(NUMJCL) |
//*          *-----*
//*
//*      1. All steps are Step restartable
//*      2. SAS program code must be saved in a library
//*          in a member named READNUM
//*      3. read in Numident raw data
//*****
//  SET REG='64M'
//*
//JS010 EXEC SAS9,
//      WORK='200000,100000', F10
//      REGION=&REG,
//      PARM='MEMSIZE=&REG'
//*
//IN1   DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI1.R110523,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*     UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*     UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*     UNIT=TSILO
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI1.SSD,
//      DISP=(OLD,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//      DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//      WORK='200000,100000',
//      REGION=&REG,
//      PARM='MEMSIZE=&REG'
//*
//IN1   DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI2.R110524,DISP=SHR
//TEMP  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*     UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
```

Appendix A.46
JCL: NUMJCL

```
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI2.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS030    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI3.R110525,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI3.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS040    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI4.R110526,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
```

Appendix A.46
JCL: NUMJCL

```
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI4.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS050    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI5.R110527,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI5.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS060    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI6.R110531,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
```

Appendix A.46
JCL: NUMJCL

```
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI6.SSD,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
// DISP=(SHR,PASS,KEEP)
//*
//JS070 EXEC SAS9,
// WORK='200000,100000',
// REGION=&REG,
// PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI7.R110601,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI7.SSD,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
// DISP=(SHR,PASS,KEEP)
//*
//JS080 EXEC SAS9,
// WORK='200000,100000',
// REGION=&REG,
// PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI8.R110602,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//TEMP1 DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//TEMP2 DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
//* UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI8.SSD,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE)
```

Appendix A.46
JCL: NUMJCL

```
//*  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),  
// DISP=(SHR,PASS,KEEP)  
//*  
//JS090 EXEC SAS9,  
// WORK='200000,100000',  
// REGION=&REG,  
// PARM='MEMSIZE=&REG'  
//*  
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI9.R110607,DISP=SHR  
//TEMP DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//TEMP1 DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//TEMP2 DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI9.SSD,  
// DISP=(OLD,CATLG,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//*  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),  
// DISP=(SHR,PASS,KEEP)  
//*  
//JS100 EXEC SAS9,  
// WORK='200000,100000',  
// REGION=&REG,  
// PARM='MEMSIZE=&REG'  
//*  
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI10.R110609,DISP=SHR  
//TEMP DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//TEMP1 DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//TEMP2 DD DSN=&&TEMP,  
// DISP=(NEW,DELETE,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//* UNIT=TSILO  
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI10.SSD,  
// DISP=(OLD,CATLG,DELETE),  
// SPACE=(CYL,(1000,100),RLSE)  
//*  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),  
// DISP=(SHR,PASS,KEEP)
```

Appendix A.46
JCL: NUMJCL

```
//*  
//JS110 EXEC SAS9,  
//      WORK='200000,100000',  
//      REGION=&REG,  
//      PARM='MEMSIZE=&REG'  
//*  
//IN1   DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI11.R110613,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//TEMP1 DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//TEMP2 DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI11.SSD,  
//      DISP=(OLD,CATLG,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS120 EXEC SAS9,  
//      WORK='200000,100000',  
//      REGION=&REG,  
//      PARM='MEMSIZE=&REG'  
//*  
//IN1   DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI12.R110615,DISP=SHR  
//TEMP  DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//TEMP1 DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//TEMP2 DD DSN=&&TEMP,  
//      DISP=(NEW,DELETE,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*    UNIT=TSILO  
//OUT1  DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI12.SSD,  
//      DISP=(OLD,CATLG,DELETE),  
//      SPACE=(CYL,(1000,100),RLSE)  
//*  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),  
//      DISP=(SHR,PASS,KEEP)  
//*  
//JS130 EXEC SAS9,  
//      WORK='200000,100000',
```

Appendix A.46
JCL: NUMJCL

```
//          REGION=&REG ,
//          PARM= 'MEMSIZE=&REG '
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI13.R110617,DISP=SHR
//TEMP     DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI13.SSD,
//          DISP=(OLD,CATLG,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM) ,
//          DISP=(SHR,PASS,KEEP)
//*
//JS140    EXEC SAS9 ,
//          WORK='200000,100000' ,
//          REGION=&REG ,
//          PARM= 'MEMSIZE=&REG '
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI14.R110621,DISP=SHR
//TEMP     DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP ,
//          DISP=(NEW,DELETE,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI14.SSD,
//          DISP=(OLD,CATLG,DELETE) ,
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM) ,
//          DISP=(SHR,PASS,KEEP)
//*
//JS150    EXEC SAS9 ,
//          WORK='200000,100000' ,
//          REGION=&REG ,
//          PARM= 'MEMSIZE=&REG '
//*
```

Appendix A.46
JCL: NUMJCL

```
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI15.R110622,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI15.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS160    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI16.R110623,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*        UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI16.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS170    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI17.R110701,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
```

Appendix A.46
JCL: NUMJCL

```
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI17.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS180    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI18.R110629,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI18.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS190    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI19.R110630,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
```

Appendix A.46
JCL: NUMJCL

```
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI19.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS200    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI20.R110705,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI20.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS210    EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI21.R110706,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
```

Appendix A.46
JCL: NUMJCL

```
//TEMP2 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI21.SSD,
//      DISP=(OLD,CATLG,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM) ,
//      DISP=(SHR,PASS,KEEP)
//*
//JS220 EXEC SAS9 ,
//      WORK='200000,100000' ,
//      REGION=&REG ,
//      PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI22.R110707,DISP=SHR
//TEMP DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//TEMP1 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//TEMP2 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI22.SSD,
//      DISP=(OLD,CATLG,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM) ,
//      DISP=(SHR,PASS,KEEP)
//*
//JS230 EXEC SAS9 ,
//      WORK='200000,100000' ,
//      REGION=&REG ,
//      PARM='MEMSIZE=&REG'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI23.R110708,DISP=SHR
//TEMP DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//TEMP1 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
//*      UNIT=TSILO
//TEMP2 DD DSN=&&TEMP ,
//      DISP=(NEW,DELETE,DELETE) ,
//      SPACE=(CYL,(1000,100),RLSE)
```

Appendix A.46
JCL: NUMJCL

```
//*          UNIT=TSILO
//OUT1      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI23.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
//JS240     EXEC SAS9,
//          WORK='200000,100000',
//          REGION=&REG,
//          PARM='MEMSIZE=&REG'
//*
//IN1       DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10.NUMI24.R110711,DISP=SHR
//TEMP      DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP1     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//TEMP2     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*          UNIT=TSILO
//OUT1      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI24.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUM.PRDLIB(READNUM),
//          DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.47
SAS Code: READNUM

```
OPTIONS NOCENTER COMPRESS=YES OBS=MAX;
/* READ IN NUMIDENT DATA */
DATA TEMP.NUM;
INFILE IN1 TRUNCOVER;
INPUT @057 REC $1.
@;
/* READ DATA FROM DIFFERENT RECORD TYPES */
IF REC IN ('G' 'J' 'K' 'O' '1' '2' '4' '5' '9'
           'P' 'S' 'W' 'V') THEN DO;
    INPUT @43  SSN      $9.
          @57  ENTCDD  $1.
          @59  CYCDTE  $8.
          @67  FRSTNAM  $15.
          @83  MIDNAM  $15.
          @99  LASTNAM  $20.
          @120 NAMESUF  $4.
          @127 DOBR    $8.
          @135 SEXR    $1.
          @136 RACER   $1.
    ;
END;
ELSE IF REC IN ('D' 'L') THEN DO;
    INPUT @43  SSN      $9.
          @57  ENTCDD  $1.
          @59  CYCDTE  $8.
          @67  FRSTNAM  $15.
          @83  MIDNAM  $15.
          @99  LASTNAM  $20.
          @120 NAMESUF  $4.
          @127 DOBR    $8.
          @135 SEXR    $1.
          @328 DODR    $8.
    ;
END;
ELSE IF REC = 'T' THEN DO;
    INPUT @43  SSN      $9.
          @57  ENTCDD  $1.
          @59  CYCDTE  $8.
          @67  FRSTNAM  $15.
          @83  MIDNAM  $15.
          @99  LASTNAM  $20.
          @120 NAMESUF  $4.
          @127 DOBR    $8.
          @135 SEXR    $1.
          @175 DODR    $8.
    ;
END;
ELSE IF REC = 'E' THEN DO;
    INPUT @43  SSN      $9.
          @57  ENTCDD  $1.
          @59  CYCDTE  $8.
          @67  FRSTNAM  $15.
          @83  MIDNAM  $15.
```

Appendix A.47
SAS Code: READNUM

```

    @99  LASTNAM $20.
    @120 NAMESUF $4.
    @127 DOBR   $8.
;
END;

IF SSN NE ' ';
RUN;
/* ORDER THE RECORDS CHRONOLOGICALLY */
PROC SORT DATA=TEMP.NUM OUT=TEMP1.NUM; BY SSN CYCDTE; RUN;
PROC PRINT DATA=TEMP1.NUM (OBS=100); TITLE 'AFTER SORT';
RUN;
DATA OUT1.NUMIDENT (KEEP=SSN
                    ENTLST
                    CYCDTLST
                    DOB
                    SEX
                    RACE
                    FRSTNAME
                    MIDNAME
                    LASTNAME
                    NAMESUFX
                    DOD )
;
SET TEMP1.NUM; BY SSN CYCDTE;

/* DATA IS ORDERED BY CYCLE DATE FOR EACH SSN          */
/* READ THROUGH ALL RECORDS AND KEEP THE NON-BLANKS     */

LENGTH
    ENTLST   $1
    CYCDTLST $8
    DOB      $8
    SEX      $1
    RACE     $1
    FRSTNAME $15
    MIDNAME  $15
    LASTNAME $20
    NAMESUFX $4
    DOD      $8
;
RETAIN
    ENTLST
    CYCDTLST
    DOB
    SEX
    RACE
    FRSTNAME
    MIDNAME
    LASTNAME
    NAMESUFX
    DOD
;
```

Appendix A.47
SAS Code: READNUM

```
ARRAY ALL(*) ENTLST
          CYCDTLST
          DOB
          SEX
          RACE
          FRSTNAME
          MIDNAME
          LASTNAME
          NAMESUFFIX
          DOD;
IF FIRST.SSN THEN DO;
  DO I = 1 TO DIM(ALL);
    ALL(I) = ' ';
  END;
END;
IF DOBR NE ' ' THEN DOB = DOBR;
IF DODR NE ' ' THEN DOD = DODR;
IF SEXR NE ' ' THEN SEX = SEXR;
IF RACER GT '0' THEN RACE = RACER;
ENTLST = ENTCD;
CYCDTLST = CYCDTE;
IF FRSTNAM NE ' ' THEN FRSTNAME = FRSTNAM;
IF MIDNAM NE ' ' THEN MIDNAME = MIDNAM;
IF LASTNAM NE ' ' THEN LASTNAME = LASTNAM;
IF NAMESUF NE ' ' THEN NAMESUFFIX = NAMESUF;
IF LAST.SSN;
RUN;
PROC PRINT DATA=OUT1.NUMIDENT (OBS=50); TITLE 'AFTER LAST'; RUN;
PROC FREQ DATA=OUT1.NUMIDENT; TABLES ENTLST; RUN;
```

Appendix A.48
JCL/SAS Code: COMBNUM

```

//#2127COM JOB (12510000,T715,,SAS,,ITC9FL),2127JLP,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          |  OPDR.TG.PRD.ETTW.N8043.LIB10(COMBNUM)  |
//*          *-----*
//*
//*          1.  COMBINES NUMIDENT FILES FOR ONE MASTER NUMIDENT
//*              CONTACT MIRIAM LOEWENBERG
//*              SSA PHONE: 202 358-6214 MPR PHONE: 202 484-4829
//*              E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//*
//*****
//  SET REG='64M'
//*
//JS010 EXEC SAS9,
//        WORK='200000,100000',
//        REGION=&REG,
//        PARM='MEMSIZE=&REG'
//*
//IN1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI1.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI2.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI3.SSD,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI4.SSD,DISP=SHR
//IN5     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI5.SSD,DISP=SHR
//IN6     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI6.SSD,DISP=SHR
//IN7     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI7.SSD,DISP=SHR
//IN8     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI8.SSD,DISP=SHR
//IN9     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI9.SSD,DISP=SHR
//IN10    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI10.SSD,DISP=SHR
//IN11    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI11.SSD,DISP=SHR
//IN12    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI12.SSD,DISP=SHR
//IN13    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI13.SSD,DISP=SHR
//IN14    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI14.SSD,DISP=SHR
//IN15    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI15.SSD,DISP=SHR
//IN16    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI16.SSD,DISP=SHR
//IN17    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI17.SSD,DISP=SHR
//IN18    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI18.SSD,DISP=SHR
//IN19    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI19.SSD,DISP=SHR
//IN20    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI20.SSD,DISP=SHR
//IN21    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI21.SSD,DISP=SHR
//IN22    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI22.SSD,DISP=SHR
//IN23    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI23.SSD,DISP=SHR
//IN24    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.NUMI24.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//OUT1    DD DSN=OPDR.TG.PRD.ETTW.#2127.COMBNUM.D1012.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,15)
//*
//SYSIN   DD *

```

Appendix A.48
JCL/SAS Code: COMBNUM

```
OPTIONS NOCENTER COMPRESS=YES OBS=MAX;
DATA TEMP.NUMIDENT;
  SET IN1.NUMIDENT  IN2.NUMIDENT  IN3.NUMIDENT  IN4.NUMIDENT
      IN5.NUMIDENT  IN6.NUMIDENT  IN7.NUMIDENT  IN8.NUMIDENT
      IN9.NUMIDENT  IN10.NUMIDENT IN11.NUMIDENT IN12.NUMIDENT
      IN13.NUMIDENT IN14.NUMIDENT IN15.NUMIDENT IN16.NUMIDENT
      IN17.NUMIDENT IN18.NUMIDENT IN19.NUMIDENT IN20.NUMIDENT
      IN21.NUMIDENT IN22.NUMIDENT IN23.NUMIDENT IN24.NUMIDENT;
  BY SSN;
RUN;

PROC SORT DATA=TEMP.NUMIDENT OUT=OUT1.NUMIDENT NODUPKEY;
  BY SSN;
RUN;

PROC CONTENTS DATA=OUT1.NUMIDENT; RUN;
PROC PRINT DATA=OUT1.NUMIDENT (OBS=25); RUN;
```

Appendix A.49
JCL/SAS Code: PAYSC09

```

//#3590PSC JOB (12510000,T715,,SAS,,ITC9FL),3590BRON,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
//*
//*      *-----*
//*      |   OPDR.TG.PRD.ETTW.#3590.TRF10.DEMO.PRDLIB(PAYSC09)
//*      *-----*
//*
//* READ IN DEC 2010 SORD DATA - PICK UP PAYMENT STATE CODE FOR DEMO
//*****
//*
//JS010    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1      DD DSN=MTOSI.CER100.FIELD.D1012,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16FIND.D1012.FLAT,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP2    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#3590.PAYSC10.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE)
//*
//SYSIN    DD *
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=YES;

/* THIS PROGRAM GETS THE STATE PAYMENT CODE FOR DECEMBER FOR
   VALIDATION TABLES - TO ADD TO DEMO */

DATA SSI;
  INFILE IN2 TRUNCOVER;
  INPUT @001 SSN $9.;
RUN;

DATA TEMP.SORD;
  INFILE IN1 TRUNCOVER;
INPUT
@0004 MFT $2.
@0015 SSN          $9.          /* PAN SOCIAL SECURITY NUMBER */
@0108 RCD_EST     ?? YMMDD8.    /* RCD-EST-JD RECORD ESTABLISHMENT
                                DATE */
@0767 PAYSC1012   $2.          /* STATE PAYMENT INDICATOR */
;
IF SUBSTR(MFT,1,1) = 'X' THEN DELETE;
/* NORA'S CONVERSATION WITH M. BARBOUR */

```

Appendix A.49
JCL/SAS Code: PAYSC09

```
IF SSN GT '000000000';

RUN;

PROC SORT DATA=TEMP.SORD OUT=TEMP1.SORD;
  BY SSN RCD_EST;
RUN;

DATA TEMP2.SORD;
  SET TEMP1.SORD;
  BY SSN RCD_EST;
  IF LAST.SSN;
RUN;
DATA OUT1.PAYSC10;
  MERGE TEMP2.SORD (IN=T) SSI (IN=S);
  BY SSN;
  IF T AND S;
RUN;
PROC FREQ DATA=OUT1.PAYSC10; TABLES PAYSC1012;
PROC CONTENTS DATA=OUT1.PAYSC10; RUN;
PROC PRINT DATA=OUT1.PAYSC10 (OBS=25); RUN;
```

Appendix A.50
JCL/SAS Code: DCFMEDEX

```
// $2358DCF JOB (12510000,T715,,SAS,,ITC9TH),2358HAZE,CLASS=T,
// MSGCLASS=1,MSGLEVEL=(1,1),REGION=4096K,NOTIFY=$2358
/*SSAMAIN SASBASIC
//*****
/* PROJECT: BUILD TRF10 (06979)
/* PROJECT
/* DIRECTOR: LAURA KOSAR
/*
/* PROGRAM: OPDR.TG.PRD.ETTW.$2358.TRF10.DEMO.PRDLIB(DCFMEDEX)
/*
/* DESCRIPTION: CREATES EXTRACT OF DCF MEDICAL FILE
/* TO OBTAIN THE BLINDESS ONEST DATE FOR THE DEMO
/*
/* DATE: 06/11/10 DAWN PHELPS
/* UPDATED: 12/1/11 NATALIE HAZELWOOD
//*****
//S1 EXEC SAS9,LOAD='DBP8.DB2.SDSNLOAD',SORT='15000',
// WORK='15000,15000',COND=EVEN
//MED DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFMED.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=TSILO,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HMED DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHMED.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=MED,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//SYSIN DD *

options nocenter;

LIBNAME MED TAPE;
LIBNAME HMED TAPE;
LIBNAME MDCF DB2 SSID=DBP8 SCHEMA=MDCF ;

* CREATE SAS EXTRACT OF CLAIM MEDICAL DATA;
PROC SQL;
CREATE TABLE MED.MED AS
SELECT * FROM MDCF.MEDICAL;
QUIT;
RUN;

* CREATE SAS EXTRACT OF HISTORY MEDICAL GROUP DATA;
PROC SQL;
CREATE TABLE HMED.HMED AS
SELECT * FROM MDCF.HMEDICAL;
QUIT;
RUN;

PROC CONTENTS DATA=MED.MED;
RUN;
PROC CONTENTS DATA=HMED.HMED;
```

Appendix A.50
JCL/SAS Code: DCFMEDEX

RUN;

Appendix A.51
JCL/SAS Code: DCFCLMEX

```
// $2358DCF JOB (12510000,T715,,SAS,,ITC9TH),2358HAZE,CLASS=T,
// MSGCLASS=1,MSGLEVEL=(1,1),REGION=4096K,NOTIFY=$2358
/*SSAMAIN SASBASIC
//*****
/* PROJECT: BUILD TRF10 (06979)
/* PROJECT DIRECTOR: LAURA KOSAR
/* PROGRAM: OPDR.TG.PRD.ETTW.$2358.TRF10.DCFDEM.PRDLIB(DCFCLMEX)
/*
/* DESCRIPTION: CREATES EXTRACT OF DCF CLAIM FILE
/* TO OBTAIN THE TWP COMPLETION MONTHS FOR DEMO
/*
/* DATE: 06/11/10 DAWN PHELPS
/* UPDATED: 12/1/11 NATALIE HAZELWOOD
//*****
//S1 EXEC SAS9,LOAD='DBP8.DB2.SDSNLOAD',SORT='15000',
// WORK='15000,15000',COND=EVEN
//CLM DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFCLM.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=TSILO,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HCLM DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHCLM.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=CLM,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//SYSIN DD *
```

options nocenter;

```
LIBNAME CLM TAPE;
LIBNAME HCLM TAPE;
LIBNAME MDCF DB2 SSID=DBP8 SCHEMA=MDCF ;
```

```
* CREATE SAS EXTRACT OF CLAIM GROUP DATA;
PROC SQL;
CREATE TABLE CLM.CLM AS
SELECT * FROM MDCF.CLM;
QUIT;
RUN;
```

```
* CREATE SAS EXTRACT OF HISTORY CLAIM GROUP DATA;
PROC SQL;
CREATE TABLE HCLM.HCLM AS
SELECT * FROM MDCF.HCLM;
QUIT;
RUN;
```

```
PROC CONTENTS DATA=CLM.CLM ;
RUN;
PROC CONTENTS DATA=HCLM.HCLM ;
RUN;
```

Appendix A.52
JCL/SAS Code: TWPCMPL

```
// $2358TWP JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:    LAURA KOSAR
//*
//* PROGRAM:     OPDR.TG.PRD.ETTW.$2358.TRF10.DEMO.PRDLIB(TWPCMPL)
//*
//* DESCRIPTION: GET TWP_CMPL_MNTH VARIABLE
//*
//* DATE:        06/11/10 DAWN PHELPS
//* UPDATED:     12/1/11 NATALIE HAZELWOOD
//*****
//*
//SAS      EXEC SAS9,
//          WORK='120000,60000'
//*
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFCLM.SA.V1,DISP=SHR
//DEMO     DD DSN=OPDR.TG.PRD.ETTW.N8043.MASTFIND.D1012.SSD,DISP=SHR
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TWPCMPL.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//TEMP1    DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,,10)
//TEMP2    DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,,10)
//SYSIN    DD *

OPTIONS OBS=MAX;

* MERGE DCF CLAIM FILE WITH DEMO SSN LINK FILE TO ELIMINATE SSNS
  THAT ARE NOT IN THE TRF;
PROC SORT DATA=IN1.CLM(KEEP=COSSN TWP_CMPL_MDT) OUT=TEMP1.CLM;
  BY COSSN;
RUN;

DATA TEMP1.TRFCLAIM;
  MERGE TEMP1.CLM(IN=CLAIM)
        DEMO.MASTFIND(IN=DEMO KEEP=SSN RENAME=(SSN=COSSN))
  ;
  BY COSSN;
  IF CLAIM AND DEMO;
RUN;

PROC SORT DATA=TEMP1.TRFCLAIM;
  BY COSSN TWP_CMPL_MDT;
RUN;

DATA TEMP1.TRFCLAIM;
  SET TEMP1.TRFCLAIM;
  BY COSSN TWP_CMPL_MDT;
```

Appendix A.52
JCL/SAS Code: TWPCMPL

```
IF FIRST.TWP_CMPL_MDT THEN OUTPUT;
RUN;

PROC          TRANSPOSE          DATA=TEMP1.TRFCLAIM          OUT=TEMP2.TWPCMPL
PREFIX=TWPCMPLMNTNTH;
  BY COSSN;
  VAR TWP_CMPL_MDT;
RUN;

DATA OUT.TWPCMPL;
  SET TEMP2.TWPCMPL;
  BY COSSN;
  IF TWPCMPLMNTNTH1=. THEN DO;
    TWPCMPLMNTNTH1=TWPCMPLMNTNTH2;
    TWPCMPLMNTNTH2=TWPCMPLMNTNTH3;
    TWPCMPLMNTNTH3=TWPCMPLMNTNTH4;
    TWPCMPLMNTNTH4=TWPCMPLMNTNTH5;
    TWPCMPLMNTNTH5=. ;
  END;
RUN;

PROC CONTENTS DATA=OUT.TWPCMPL;
RUN;

DATA TEMP2.TWPTTEST;
  SET OUT.TWPCMPL;
  WHERE TWPCMPLMNTNTH3 NE .;
RUN;

PROC PRINT DATA=TEMP2.TWPTTEST(OBS=1000);
RUN;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
//#3590T2 JOB (12510000,T715,,SAS,,ITC9FL),3590XXX,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590  
//JOB01 EXEC SAS9,  
//          WORK='180000,90000'  
//*****  
//*  
//*          *-----*  
//*          | OPDR.TG.PRD.ETTW.#3590.TRF10.DEMO.PRDLIB(T2T16DEM)  
//*          *-----*  
//*  
//*****  
//*  
//DEM DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.DEMO.SA.V2,DISP=SHR  
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR  
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR  
//IN3 DD DSN=OPDR.TG.PRD.ETTW.#2127.COMBNUM.D1012.SSD,DISP=SHR  
//IN4 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR  
//IN5 DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBSORD.D1012.SSD,DISP=SHR  
//FIN DD DSN=OPDR.TG.PRD.ETTW.N8043.MASTFIND.D1012.SSD,DISP=SHR  
//TWP DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TWPCMPL.SA.V1,DISP=SHR  
//BLD DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFMED.SA.V1,DISP=SHR  
//PAY DD DSN=OPDR.TG.PRD.ETTW.#3590.PAYSC10.SSD,DISP=SHR  
//TEMP DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
//TEMP1 DD DSN=&&TEMP,  
//          DISP=(NEW,DELETE,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)  
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DEMO.SA.V1,  
//          DISP=(NEW,CATLG,DELETE),  
//          UNIT=TSILO  
//OUT2 DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,  
//          DISP=(NEW,CATLG,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE)  
//OUT3 DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,  
//          DISP=(NEW,CATLG,DELETE),  
//          SPACE=(CYL,(1000,100),RLSE)  
//SYSIN DD *
```

```
/*  
*  
* FILENAME: T2T16DEM  
* PROGRAMMER:MIRIAM LOEWENBERG  
* PURPOSE:TO CREATE DEMOGRAPHIC VARIABLES FILE FOR TRF  
* CREATED:8/11/22  
* REVISED TO DO EVERYTHING IN ONE STEP  
* BUILD NEW DATE OFF DEATH VARIABLE  
*****/  
options nocenter ls=132 ps=60 compress=YES;  
%let begyr=1994;  
%let endyr=2010; /* change as needed */  
%let endmn=12; /* change as needed */  
/* step to assign macro variables to handle time series data */
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;

%LET TOT = %EVAL(&K-1);

/* MACRO VAR TO SET STARTING POINT FOR SORD PROCESSING. WE ONLY
PROCESS THE CURRENT SORD YEAR IN EACH ITERATION OF TRF.
ADJUST THIS EACH NEW ITERATION */
%LET SORDSTART=193;

DATA TEMP.BLIND;
  SET BLD.MED;
  IF BLND_ONST_DT GT .Z;
RUN;
PROC SORT DATA=TEMP.BLIND NODUPKEY OUT=TEMP1.BLIND; BY COSSN; RUN;
DATA BLIND ;
  MERGE
    TEMP1.BLIND (IN=M RENAME=(COSSN=SSN BLND_ONST_DT=BLINDDT))
    FIN.MASTFIND (IN=L);
  BY SSN;
  IF NOT (M AND L) THEN DELETE;
RUN;

DATA OUT1.DEMO OUT2.LINKSSN(KEEP=SSN)
  OUT3.DMLDW(KEEP = SSN CAN DOBBEST DODBEST TWPCPLMNT1);
MERGE
  DEM.DEMO (IN=IND RENAME=(RACE=ORACE SEX=OSEX))
  FIN.MASTFIND (IN=MAST)
  IN1.LONG(IN = INL
  KEEP = SSN HUN TOA COMP_STA MFT AGE18_CDR
  START_RD RCD_EST BIRTH_JD DEATH_JD LAF CURSTAT SEX ELG_RD
  APP_RD _8080_JD CLM_FIL DISPAYCD DIB_DIG DIB_DIG2 DIB_MDR
  DIB_DPM PDSCC PDZIP PDZIP6_9 DO REPPAYTP FIRST_PA STOP_RD
  MAX_80JD MAXAPPRD MAXCLMFL MAXELGRD MAXFRSTP MAXRCDST
  MAXSTPRD MAXSTRRD AGE18REDDT LANG_WR
  MIN_80JD MINAPPRD MINCLMFL MINELGRD MINFRSTP MINRCDST
  MINSTPRD MINSTRRD
  T16APPL: T16START: T16STOP:
  DEC: T16RID:
  RENAME=(SEX=LONGSEX AGE18REDDT=AGE18REDDT LANG_WR=T16LANG
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```

        BIRTH_JD=DOBLONG DO=DOC ))
IN2.MBR (IN=INM KEEP=SSN CAN BGN BIC TAC DUALELIG BLN COUNTY DDO1-
DDO12

        DIG1-DIG12 SDIG1-SDIG12 ENTDAT1-ENTDAT12 DDBC1-DDBC12
        APS1-APS12 BDC1-BDC12 ADC1-ADC12 HDD1-HDD12 CSA1-CSA12
        CDR1-CDR12 DSD1-DSD12 DAC1-DAC12 LOD1-LOD12 SDS1-SDS12
        EBD1-EBD12 DAC1-DAC12 TOC1-TOC20 TOC_NUM
        TOC_START1-TOC_START20
        BDOF1-BDOF25 NDOF LANG /* NEW VARIABLES */
        DDOD DACD LODD ENTDATD DBCD DSDD HDDD SDDD ADCD APSD EBDD
        BDCD CDRD CSAD DIGD SDIGD DOEID DOECD
        NPJA NODF LSAP LSDC LSFJ LSSC: LSPA SIFT SISC
        DOB DOC LAF PIA1-PIA50 PNOB RDD RP RZIP CEC SLAC
        SEX STATE TOP PIARFC1-PIARFC50 IME1-IME50
        DOEI DOEC DOST HI_START HI_TERM PIED1-PIED50
        SMI_STAR SMI_TERM BDOD HBIC1-HBIC25 BDOE_START1-BDOE_START25
        BDOE_TERM1-BDOE_TERM25
        RENAME=(SEX=MBRSEX
        DOB=DOBMBR))
IN3.NUMIDENT(IN=INU KEEP=SSN DOD FRSTNAME LASTNAME NAMESUFIX DOB
        SEX RACE RENAME=(DOB=DOBNUM SEX=NSEX RACE=NRACE))
IN4.ALLCDR (IN=INC KEEP=SSN CDRDTE: JUDLV: RDT: RID: RB: DPM:
        %DO I=1 %TO &TOT;
        MIEX&&X&I /* FOR ESTABLISHING FIRST MIE */
        %END;)
IN5.SORD(IN=INR KEEP=PAN BIRTH_JD SEX
        %DO I=&SORDSTART %TO &TOT; /* START WITH JAN OF TRF YEAR*/
        DX&&X&I CIT&&X&I
        %END;
        RENAME=(BIRTH_JD=DOBREM PAN=SSN SEX=RSEX
        %DO I=&SORDSTART %TO &TOT;
        DX&&X&I=RD&&X&I
        %END;))
TWP.TWPCMPL(IN=INTW KEEP=COSSN TWPCMPLMNTN:
        RENAME=(COSSN=SSN))
BLIND(IN=INBL KEEP=SSN BLINDDT)
PAY.PAYSC10(IN=INPY KEEP=SSN PAYSC1012
        RENAME=(PAYSC1012=PAYSC_DEC10))
;
        BY SSN;
/* INDICATORS FOR MERGING */

        DEMOIND = 0; SSIIND = 0; SSDIIND = 0; CDRIND = 0;
        IF IND THEN DEMOIND = 1;
        IF INL THEN SSIIND = 1;
        IF INM THEN SSDIIND = 1;
        IF INC THEN CDRIND = 1;

/* SUBSET TO SSN'S IN THE MASTER FINDER FILE */
IF NOT MAST THEN DELETE;

/* SCREEN OUT CASES WHICH HAVE THE WRONG DATA BECAUSE 2% OF MBR

```

Appendix A.53
JCL/SAS Code: T2T16DEM

RECORDS HAVE THE WRONG BOAN. WE WILL CHECK MBR BIRTH YEAR AND FIRST NAME AGAINST NUMIDENT BIRTH YEAR AND FIRST NAME. WHERE THE FIRST NAME IS DIFFERENT AND THE BIRTH YEARS ARE MORE THEN 2 YEARS APART WE WILL DELETE THE RECORD. THE CHECK ON FIRST NAME MAKES SENSE BECAUSE THESE CASES WILL MOST OFTEN OCCUR FOR DEPENDENTS WHERE THE PRIMARY HAS THE SAME LAST NAME */

```
IF DOBMBR NE . THEN MBRYR = YEAR(DOBMBR);
IF SUBSTR(DOBNUM,7,2) NE 'XX' THEN NYR = SUBSTR(DOBNUM,5,4);
NUMYR = INPUT(NYR,8.);
```

```
DFLAG = 0;
IF MBRYR GT .Z AND NUMYR GT .Z THEN DO;
  IF BGN NE FRSTNAME AND ABS(MBRYR-NUMYR) > 2 THEN DFLAG = 1;
END;
```

```
IF DFLAG = 1 THEN DELETE;
```

```
/* CREATE THE BEST DATE OF BIRTH - DOBBEST
THE NUMIDENT DATA OF BIRTH IS A CHARACTER VARIABLE.
START WITH NUMIDENT - IF IT READS INTO A SAS DATE THEN = DOBBEST.
BUT THERE MAY BE MISSING DAYS AND MONTHS WHERE THE YEAR IS
PRESENT.
IN THIS CASE COMPARE THE YEAR TO LONGDOB, MBRDOB AND REMDOB IN
THAT
ORDER. WHERE AN EQUAL YEAR IS FOUND - FILL IN THE MONTH AND DAY.
WHERE THE ENTIRE NUMIDENT DOB IS MISSING, EXAMINE LONGDOB-MBRDOB-
REMDBO IN THAT ORDER AND USE THE FIRST VALID DATE.
AT THE END OF THE PROCESS, IF DOBBEST IS STILL MISSING BECAUSE NO
VALUES WERE FOUND IN LONGDOB, MBRDOB, AND REMDOB:
  IF THE NUMIDENT YEAR IS PRESENT, SET JAN 1 AS THE MONTH AND DAY
  AND CREATE A FLAG - DOBFLAG- TO MARK THOSE CASES. */
```

```
IF DOBLONG NE . THEN LONGYR = YEAR(DOBLONG);
IF DOBREM NE . THEN REMYR = YEAR(DOBREM);
```

```
DOBBEST = INPUT(DOBNUM,YYMMDD10.);
```

```
IF DOBBEST = . THEN DO;
```

```
IF NUMYR NE . THEN DO;
  IF LONGYR = NUMYR THEN DOBBEST = DOBLONG;
  ELSE IF MBRYR = NUMYR THEN DOBBEST = DOBMBR;
  ELSE IF REMYR = NUMYR THEN DOBBEST = DOBREM;
END;
```

```
/* IF NO MATCH IS FOUND SELECT THE FIRST VALID DOB FROM
THE THREE OTHER DATES */
```

```
ELSE IF DOBLONG NE . THEN DOBBEST = DOBLONG;
ELSE IF DOBMBR NE . THEN DOBBEST = DOBMBR;
ELSE IF DOBREM NE . THEN DOBBEST = DOBREM;
```

```
END;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
/* IF AT THE END OF THE PROCESS, NUMIDENT YEAR IS THE ONLY INFO
AVAILABLE, CREATE DUMMY MONTH AND DAY AND USE THE YEAR FOR
DOBBEST.
```

```
CREATE A FLAG FOR THESE CASES */
```

```
DOBFLAG = 0;
```

```
IF DOBBEST = . AND NUMYR NE . THEN DO;
```

```
M = 1;
```

```
D = 1;
```

```
DOBBEST = MDY(M,D,NUMYR);
```

```
DOBFLAG = 1;
```

```
END;
```

```
FORMAT DOBBEST YMMDD10.;
```

```
/* CREATE THE BEST DATE OF DEATH - DODBEST
THE NUMIDENT DATA OF DEATH IS A CHARACTER VARIABLE.
START WITH NUMIDENT - IF IT READS INTO A SAS DATE THEN = DODBEST.
BUT THERE MAY BE MISSING DAYS AND MONTHS WHERE THE YEAR IS
PRESENT.
```

```
IN THIS CASE COMPARE THE YEAR TO DEATH_JD AND BDOD IN THAT
ORDER. WHERE AN EQUAL YEAR IS FOUND - FILL IN THE MONTH AND DAY.
WHERE THE ENTIRE NUMIDENT DOD IS MISSING, EXAMINE DEATH_JD AND
BDOD IN THAT ORDER AND USE THE FIRST VALID DATE.
```

```
AT THE END OF THE PROCESS, IF DODBEST IS STILL MISSING BECAUSE NO
VALUES WERE FOUND IN DEATH_JD AND BDOD:
```

```
IF THE NUMIDENT YEAR IS PRESENT, SET JAN 1 AS THE MONTH AND DAY
AND CREATE A FLAG - DODFLAG- TO MARK THOSE CASES. */
```

```
IF SUBSTR(DOD,7,2) NE 'XX' THEN NYR = SUBSTR(DOD,5,4);
```

```
NUMYRD = INPUT(NYR,8.);
```

```
IF DEATH_JD NE . THEN LONGYRD = YEAR(DEATH_JD);
```

```
IF BDOD NE . THEN MBRYRD = YEAR(BDOD);
```

```
DODBEST = INPUT(DOD,YMMDD10.);
```

```
IF DODBEST = . THEN DO;
```

```
IF NUMYRD NE . THEN DO;
```

```
IF LONGYRD = NUMYRD THEN DODBEST = DEATH_JD;
```

```
ELSE IF MBRYRD = NUMYRD THEN DODBEST = BDOD;
```

```
END;
```

```
/* IF NO MATCH IS FOUND SELECT THE FIRST VALID DOD FROM
THE TWO OTHER DATES */
```

```
ELSE IF DEATH_JD NE . THEN DODBEST = DEATH_JD;
```

```
ELSE IF BDOD NE . THEN DODBEST = BDOD;
```

```
END;
```

```
/* IF AT THE END OF THE PROCESS, NUMIDENT YEAR IS THE ONLY INFO
AVAILABLE, CREATE DUMMY MONTH AND DAY AND USE THE YEAR FOR
DODBEST.
```

```
CREATE A FLAG FOR THESE CASES */
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
DODFLAG = 0;

IF DODBEST = . AND NUMYRD NE . THEN DO;
  M = 1;
  D = 1;
  DODBEST = MDY(M,D,NUMYRD);
  DODFLAG = 1;
END;
FORMAT DODBEST YMMDD10.;

/* ADD BLIND DATE */
/*
BLINDDT=INPUT(BLNDDT,YMMDD10.);
DROP BLNDDT;
*/
/* CREATE JUDGEMENT INDICATOR (FROM DODEC) VARIABLES FROM 831 DATA
*/

ARRAY CDR(*) CDRDTE1-CDRDTE48;
ARRAY DOD1(*) DODEC1-DODEC48; /* RENAME CDRDTEn VARS */
ARRAY LEV(*) JUDLVL1-JUDLVL48;
ARRAY RDT(*) RDT1-RDT48;
ARRAY RID(*) RID1-RID48;

FRST_DODEC = CDRDTE1;
FORMAT FRST_DODEC YMMDD10.;

DO I = 1 TO DIM(CDR);
  DOD1(I) = CDR(I);
  IF CDR(I) GT .Z THEN DO;
    LAST_DODEC = CDR(I);
  END;
END;

/* ESTABLISH THE DATE OF THE FIRST MIEX */
/* USE THE VNAME FUNCTION TO CREATE THE VALUE FOR YEAR AND MONTH */

ARRAY MIEX{*} $ %DO I=1 %TO &TOT; MIEX&&X&I %END;;

LENGTH MIENAME $8;
LENGTH MIE_BEFORE_CDR $1.;

MIENAME = ' ';
/* INITIALIZE INDICATOR VARIABLE FOR MIE TO BLANK. IF THERE IS A
RECORD IN THE ALLCDR FILE THEN INITIALIZE TO 'N'. */

MIE_BEFORE_CDR = ' ';
IF INC THEN MIE_BEFORE_CDR = 'N';

DO I=1 TO DIM(MIEX) UNTIL (MIENAME NE ' ');
  /* E MEANS MEDICAL IMPROVEMENT EXPECTED */
  IF MIEX(I) EQ 'E' THEN MIENAME = VNAME(MIEX(I));
END;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
YR = SUBSTR(MIENAME,5,2);
  IF YR LT '90' THEN YR = '20' || YR;
  ELSE YR = '19' || YR;
YEAR = INPUT(YR,8.);
MONTH = INPUT(SUBSTR(MIENAME,7,2),8.);
DAY = 1;
FRST_MIE = MDY(MONTH,DAY,YEAR);
FORMAT FRST_MIE YMMDD10.;

/* DETERMINE IF MIE BEFORE THE FIRST CONTINUING DISABILITY REVIEW.
  THE MIE MONTHS ARE SET FROM THE DATE OF ENTITLEMENT TO DISABILITY
  OR THE DODEC FIELD IF THE FIRST IS MISSING. THE DATE OF
ENTITLEMENT
  IS SET AFTER THE JUDGMENT AFFIRMS DISABILITY AND IS MOST OFTEN
  EARLIER THAN THE JUDGMENT DATE SINCE IT TAKES TIME FOR A
  JUDGMENT TO BE MADE. THEREFORE WE CHECK WHETHER THE FIRST MIE
  OCCURS EARLIER OR ON THE SAME DATE AS ANY ADJUDICATION EVENT
  ON THE 833 FILE. THERE MAY BE MANY DENIALS BEFORE A CLAIM IS
  APPROVED SO WE NEED TO CHECK THE RESULT OF DETERMINATION FIELD
  (RDT). THE DAY PORTION OF THE FIRST MIE FIELD IS
  SET TO "1" SO IT WILL ALWAYS BE LESS THAN OR = TO CDRDTEN.
  BUT WE HAVE MONTHLY MIE'S ONLY FROM 1994 - IF THE FIRST DODEC
  IS BEFORE THEN WE ARE LIMITED IN ESTABLISHING MIE BEFORE CDR. */
IF FRST_MIE NE . THEN DO;
  /* STOP LOOPING WHEN AN 833 EVENT OCCURS */
  DO I = 1 TO DIM(CDR) WHILE (RID(I) IN ('2' 'R'));
  IF FRST_MIE LE CDR(I) AND RDT(I) = 'A' THEN MIE_BEFORE_CDR =
'Y';
  /* IF MIE DATE IS JAN 1 1994 WE CANNOT DETERMINE ITS
  RELATIONSHIP TO A CDR DATE AND SO WE SET THE VARIABLE TO
  MISSING */
  IF FRST_MIE EQ '01JAN1994'D THEN MIE_BEFORE_CDR = ' ';
END;
END;

DROP MIENAME YR YEAR MONTH DAY;
/* CREATE REPRESENTATIVE PAYEE INDICATOR FOR SSI */

IF REPPAYTP IN (' ','SEL') THEN REPPYSSI = 0;
ELSE IF REPPAYTP > ' ' THEN REPPYSSI = 1;
ELSE REPPYSSI = .;

/* CREATE REPRESENTATIVE PAYEE INDICATOR */

IF TOP IN (' ','A') THEN REPPYSSD = 0;
ELSE IF TOP > ' ' THEN REPPYSSD = 1;
ELSE REPPYSSD = .;

/* TRIAL WORK PERIOD INDICATOR MOVED TO DCF */

IF MAXSTRRD = . AND MAXELGRD NE . THEN MAXSTRRD = MAXELGRD;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
ELSE IF MAXSTRRD = . AND MAXELGRD = . THEN MAXSTRRD = .;

/* PROCESS SORD VARIABLES */
/* ONLY FOR CURRENT ONGOING AND NEW CASES */
IF INR THEN DO;

  ARRAY REMDX{*} $ %DO I=&SORDSTART %TO &TOT; RDX&&X&I %END;;
  ARRAY REMCT{*} $ %DO I=&SORDSTART %TO &TOT; CIT&&X&I %END;;

  DO I=1 TO DIM(REMDX);
    IF REMDX{I} NE ' ' THEN REMDIG1N=REMDX{I};
    IF REMCT{I} NE ' ' THEN CITIZENN=REMCT{I};
  END;
  DROP %DO I=&SORDSTART %TO &TOT; RDX&&X&I CIT&&X&I %END;;
END; /* INR */

/* SET THE SORD VARIABLES - DO NOT OVERWRITE WITH BLANK */
IF CITIZEN = ' ' THEN CITIZEN = CITIZENN;
IF REMDIG1 = ' ' THEN REMDIG1 = REMDIG1N;
REMSEX = RSEX;
DROP CITIZENN REMDIG1N ;

/* TAKE SEX VALUES FROM NUMIDENT FIRST. TAKE RACE FROM NUMIDENT.
NOT DONE FOR TRF1. THESE VALUES SHOULD BE THE MOST ACCURATE.
RAND DOCUMENTATION SAYS THAT RACE NOT RELIABLE IN SSR AND
FOR MBR IT IS THE RACE OF THE PRIMARY - NOT ALWAYS APPROPRIATE
FOR DAC'S AND WIDOWS - REVISE THE SPI CODE */

/* RECODE NUMIDENT SEX */

IF NSEX = '1' THEN NUM_SEX = 'M';
ELSE IF NSEX = '2' THEN NUM_SEX = 'F';

IF NSEX NE ' ' THEN SEX = NUM_SEX;
ELSE IF SEX = ' ' THEN DO;

  /* FIX SEX FOR PEOPLE WHOSE VALUES DONT MATCH ACROSS EXTRACTS */

  IF LONGSEX = ' ' AND MBRSEX = ' ' AND REMSEX > ' ' THEN SEX = REMSEX;
  ELSE IF LONGSEX = ' ' AND MBRSEX = ' ' AND REMSEX = ' ' THEN SEX =
LONGSEX;
  ELSE IF LONGSEX = ' ' AND MBRSEX > ' ' AND REMSEX = ' ' THEN SEX =
MBRSEX;
  ELSE IF LONGSEX = ' ' AND MBRSEX > ' ' AND REMSEX > ' ' THEN
SEX = MBRSEX;
  ELSE IF LONGSEX > ' ' AND REMSEX > ' ' AND MBRSEX = ' ' THEN
SEX = LONGSEX;
  ELSE IF LONGSEX > ' ' AND REMSEX = ' ' AND MBRSEX > ' ' THEN DO;
  IF MAX(MAXSTRRD,DOEI,DOEC) = DOEI OR
      MAX(MAXSTRRD,DOEI,DOEC) = DOEC THEN SEX = MBRSEX;
  ELSE IF MAX(MAXSTRRD,DOEI,DOEC) = MAXSTRRD THEN SEX = LONGSEX;
  END;
  ELSE IF LONGSEX > ' ' AND REMSEX > ' ' AND MBRSEX > ' ' THEN DO;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
IF MAX(MAXSTRRD,DOEI,DOEC) = DOEI OR
    MAX(MAXSTRRD,DOEI,DOEC) = DOEC THEN SEX = MBRSEX;
ELSE IF MAX(MAXSTRRD,DOEI,DOEC) = MAXSTRRD THEN SEX = LONGSEX;
END;
ELSE SEX = MBRSEX;
END; /* ELSE DO - NOT IN NUMIDENT OR OLD FILE */

/* IF SEX WAS NOT OVERWRITTEN BY NEW DATA THEN USE OLD VALUE */
IF OSEX NE ' ' AND SEX = ' ' THEN SEX = OSEX;

DROP LONGSEX MBRSEX REMSEX NSEX NUM_SEX OSEX;
/* CREATE INDICATOR VARIABLE FOR MALE */

IF SEX = 'M' THEN MALE = 1;
ELSE IF SEX = 'F' THEN MALE = 0;
ELSE MALE = .;
SEXMISS = 0;
IF SEX = ' ' THEN SEXMISS = 1;

/* FOR RACE USE NUMIDENT */
/* RECODE NUMIDENT RACE VALUES */
IF NRACE = '0' THEN NUM_RACE = 'U';
ELSE IF NRACE = '1' THEN NUM_RACE = 'W';
ELSE IF NRACE = '2' THEN NUM_RACE = 'B';
ELSE IF NRACE = '3' THEN NUM_RACE = 'O';
ELSE IF NRACE = '4' THEN NUM_RACE = 'A';
ELSE IF NRACE = '5' THEN NUM_RACE = 'H';
ELSE IF NRACE = '6' THEN NUM_RACE = 'I';

/* CAPTURE THE FEW RECORDS FROM OLD TRF WHO ARE NOT IN THE
CURRENT NUMIDENT FILE */
ELSE NUM_RACE = ORACE;
RACE = NUM_RACE;
DROP NRACE NUM_RACE ORACE;

/* CREATE INDICATOR VARIABLES FOR DIFFERENT RACES/ETHNICITIES */

IF RACE = ' ' THEN RACEMISS = 1;
ELSE IF RACE > ' ' THEN RACEMISS = 0;

/* CREATE BINARIES */
IF RACEMISS = 0 THEN DO;

ASIAN = 0;
IF RACE = 'A' THEN ASIAN = 1;

BLACK = 0;
IF RACE IN ('B','N') THEN BLACK = 1;

HISPANIC = 0;
IF RACE = 'H' THEN HISPANIC = 1;

NAINDIAN = 0;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
IF RACE = 'I' THEN NAINDIAN = 1;

WHITE = 0;
IF RACE = 'W' THEN WHITE = 1;

OTHER = 0;
IF RACE = 'O' THEN OTHER = 1;

UNKNOWN = 0;
IF RACE = 'U' THEN UNKNOWN = 1;

END;
LENGTH ASIAN BLACK
HISPANIC WHITE OTHER UNKNOWN
MALE NAINDIAN RACEMISS SEXMISS
3;

LENGTH
ENTDAT1-ENTDAT12 ENTDATD DOBLONG DOBREM DDO1-DDO12 DDOD DDBC1-DDBC12
DDBCD DSD1-DSD12 DSDD HDD1-HDD12 HDDD SDS1-SDS12 SDSD ADC1-ADC12 ADCD
APS1-APS12 APSD EBD1-EBD12 EBDD
DOBMBR MAXAPPRD MAXCLMFL MAXELGRD MAXFRSTP MAXRC DST
MAXSTPRD MAXSTRRD MAX_80JD MINAPPRD MINCLMFL
MINELGRD MINFRSTP MINRC DST MINSTPRD MINSTRRD MIN_80JD
4
REPPYSSI REPPYSSD 3
;

LABEL
CAN = 'T2 CLAIM ACCOUNT NUMBER'
FRSTNAME = 'NUMIDENT FIRST NAME'
LASTNAME = 'NUMIDENT LAST NAME'
NAMESUFX = 'NUMIDENT NAME SUFFIX'
DOBNUM = 'NUMIDENT DATE OF BIRTH'
NODF = 'NUMBER OF MBR DISABILITY FIELDS'
NDOF = 'NUMBER OF DATE OF FILING FIELDS'
NPJA = 'NUMBER OF MBR PIA ENTIERES'
FRST_MIE = 'DATE MIE FIRST FLAGGED'
MIE_BEFORE_CDR = 'MIE BEFORE FIRST CDR'
DOBFLAG = 'JAN 1 IMPUTED FOR BIRTH DATE'
DODFLAG = 'JAN 1 IMPUTED FOR DEATH DATE'
RP = 'T2 RACE OF THE PRIMARY'
TAC = 'TYPE OF AWARD CODE'
SISC = 'SSI INCOME STATUS CODE '
SIFT = 'SECURITY INCOME FILE TYPE'
%DO I = 1 %TO 12;
ADC&I = "T2 APPLICANTS DISABILITY CESSATION ENTRY &I"
EBD&I = "T2 EPE BEGIN DATE ENTRY &I"
%END;
%DO I = 1 %TO 25;
HBIC&I="HISTORICAL BIC &I"
BDOE_START&I = "HISTORICAL BIC START DATE &I"
BDOE_TERM&I = "HISTORICAL BIC END DATE &I"
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
%END;
%DO I = 1 %TO 25;
  BDOF&I = "T2 APPLICANTS DISABILITY CESSATION ENTRY &I"
%END;
%DO I = 1 %TO 4;
  TWPCMLMNT&I = "TRIAL WORK PERIOD COMPLETION MONTH ENTRY &I"
%END;
%DO I = 1 %TO 20;
  TOC&I = "T2 TYPE OF CLAIM"
  TOC_START&I = "T2 TOC START DATE"
%END;
%DO I = 1 %TO 48;
  DODEC&I = "831-833 DODEC (JUDGMENT DATE) ENTRY &I"
  DPM&I = "831-833 PERMANENT DISABILITY CODE ENTRY &I"
%END;
CURSTAT = 'T16 CURRENT PAYMENT STATUS'
DEATH_JD = 'T16 DEATH DATE'
DIB_DPM = 'T16 PERMANENT DISABILITY IND'
DISPAYCD = 'T16 DISABILITY PAYMENT CODE'
FIRST_PA = 'T16 1ST PAYMENT DATE (THIS RECORD)'
DOBLONG = 'T16 DATE OF BIRTH'
MFT = 'T16 MASTER FILE TYPE CODE'
PDSCC = 'T16 STATE/COUNTY/DO CODE'
PDZIP = 'T16 PAYEE ZIP CODE'
PDZIP6_9 = 'T16 PAYEE ZIP SUFFIX'
RCD_EST = 'T16 RECORD ESTABLISHMENT DATE'
REPPAYTP = 'T16 TYPE OF PAYEE CODE'
START_RD = 'T16 EARLIEST COMPUTATION DATE'
STOP_RD = 'T16 STOP DATE'
TOA = 'T16 TYPE OF ACTION'
COMP_STA = 'T16 STATUS TYPE OF ACTION'
APP_RD = 'T16 APPLICATION DATE (CURRENT)'
CLM_FIL = 'T16 DATE CLAIM FILED'
DOBMBR = 'T2-MBR BIRTH DATE'
DOBREM = 'REMICS BIRTH DATE'
ELG_RD = 'T16 DATE OF CURRENT ELIGIBILITY'
TOP = 'TYPE OF PAYEE'
_8080_JD = 'INITIAL DECISION DATE'
DOD = 'DATE OF DEATH'
BGN = 'FIRST NAME'
BIC = 'BENEFICIARY IDENTIFICATION CODE'
DIB_DIG = 'PRIMARY 4-DIGIT DIAGNOSIS (SSI LONG)'
DIB_DIG2 = 'SECONDARY 4-DIGIT DIAGNOSIS'
DIB_MDR = 'MEDICAL DIARY REASON'
BLN = 'LAST NAME'
DOC = 'DISTRICT OFFICE CODE'
DOEC = 'DATE OF CURRENT ENTITLEMENT - SSDI'
DOEI = 'DATE OF INITIAL ENTITLEMENT - SSDI'
DOST = 'DATE OF SUSPENSION/TERMINATION'
HI_START = 'START DATE FOR HI'
HI_TERM = 'END DATE FOR HI'
TOC_NUM = 'T2 NUMBER OF TOC OCCURRENCES'
LAF = 'LEDGER ACCOUNT FILE STATUS'
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
PNOB      = 'NUMBER OF BENEFICIARIES IN PAYMENT'
REPPYSSI  = 'REPRESENTATIVE PAYEE INDICATOR-T16'
REPPYSSD  = 'REPRESENTATIVE PAYEE INDICATOR-T2'

SMI_STAR  = 'START DATE FOR SMI'
SMI_TERM  = 'END DATE FOR SMI'

COUNTY   = 'SSA COUNTY CODE FOR RESIDENCE'

MAXAPPRD  = 'LATEST APPLICATION DATE'
MAXCLMFL  = 'LATEST APPLICATION RECEIPT DATE'
MAXELGRD  = 'LATEST SSI ELIGIBILITY DATE'
MAXFRSTP  = 'LATEST APPLICATION, FIRST SSI PAYMENT DT'
MAXRCDST  = 'LATEST SSR RECORD ESTABLISHMENT DATE'
MAXSTPRD  = 'LATEST APPLICATION, ELIG PERIOD END DATE'
MAXSTRRD  = 'MOST RECENT START DATE OF SSI PAYMENTS'
MAX_80JD  = 'LATEST DECISION DATE'
MINAPPRD  = 'FIRST APPLICATION DATE'
MINCLMFL  = 'FIRST APPLICATION RECEIPT DATE'
MINELGRD  = 'FIRST SSI ELIGIBILITY DATE'
MINFRSTP  = 'FIRST APPLICATION, FIRST SSI PAYMENT DT'
MINRCDST  = 'FIRST SSR RECORD ESTABLISHMENT DATE'
MINSTPRD  = 'FIRST APPLICATION, ELIG PERIOD END DATE'
MINSTRRD  = 'FIRST APPLICATION COMPUTATION DATE'
MIN_80JD  = 'FIRST DECISION DATE'
RDD       = 'REASON FOR DEDUCTION'
CEC       = 'DI CURRENT ENTITLEMENT CODE'
SLAC      = 'SSI LIVING ARRANGEMENT CODE'
LSPA      = 'DI LUMP SUM PAYMENT AMOUNT'
LSAP      = 'DI LUMP SUM AWARD AMOUNT'
LSDC      = 'DI LUMP SUM DISALLOWANCE CODE'
LSFD      = 'DI LUMP SUM FILING DATE'
LSSC_AUTPMT = 'DI LUMP SUM CODE-AUTHORIZED PAYMENT'
LSSC_BURL  = 'DI LUMP SUM CODE-BURIAL EXPENSE'
LSSC_CHLD  = 'DI LUMP SUM CODE-ENTITLED CHILD'
LSSC_DSALLW = 'DI LUMP SUM CODE-CLAIMANT DISALLOWED'
LSSC_FUNR  = 'DI LUMP SUM CODE-FUNERAL HOME'
LSSC_LVSPS = 'DI LUMP SUM CODE-LIVING WITH SPOUSE'
LSSC_WID   = 'DI LUMP SUM CODE-ENTITLED WIDOW'

%DO I=1 %TO 50;
  IME&I = 'INDEXED MONTHLY EARNINGS'
  PIA&I = 'PRIMARY INSURANCE AMOUNT'
  PIED&I = 'PIA EFFECTIVE DATE'
  PIARFC&I = 'REASON FOR CHANGE IN PIA'
%END;

RDD       = 'REASON FOR DEDUCTION'
RZIP      = 'ZIP CODE'
SSN       = 'SOCIAL SECURITY NUMBER'
STATE     = 'SSA STATE CODE FOR RESIDENCE'
TOP       = 'TYPE OF PAYEE'
;
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
LABEL
ASIAN      = 'ASIAN INDICATOR'
BLACK     = 'AFRICAN AMERICAN INDICATOR'
WHITE     = 'WHITE AMERICAN INDICATOR'
OTHER     = 'OTHER-RACE AMERICAN INDICATOR'
UNKNOWN   = 'UNKNOWN-RACE AMERICAN INDICATOR'
HISPANIC  = 'HISPANIC RACE INDICATOR'
NAINDIAN  = 'NATIVE AMERICAN INDICATOR'
CITIZEN   = 'US CITIZEN INDICATOR'
FRST_DODEC = 'FIRST ADJUDICATION DATE'
LAST_DODEC = 'LAST ADJUDICATION DATE'

HISPANIC = 'HISPANIC INDICATOR'

MALE      = 'MALE INDICATOR'

REMDIG1 = 'REMICS PRIMARY 4-DIGIT DIAGNOSIS'
RACE     = 'RACE'
RACEMISS = 'OTHER/UNKNOWN RACE INDICATOR'
SEXMISS  = 'MISSING SEX VARIABLE'
DOBBEST  = 'BEST BIRTH DATE'
DODBEST  = 'BEST DEATH DATE'
HUN      = 'SSI HOUSED UNDER NUMBER'
SEX      = 'SEX'
BDOD     = 'MBR DATE OF DEATH'
LANG     = 'MBR WRITTEN LANGUAGE INDICATOR'
T16LANG  = 'SSI LANGUAGE INDICATOR'
ERP_IND  = 'ECONOMIC RECOVERY INDICATOR MAY 09'
AGE18REDDT = 'AGE 18 REDETERMINATION DATE'
;

IF SSN GT '000000000';

DROP %DO I=1 %TO &TOT; MIEX&&X&I %END;;
DROP I;
DROP CDRDTE1-CDRDTE48 D DFLAG LONGYR LONGYRD NUMYRD MBRYRD
M MBRYR NUMYR NYR REMYR RSEX;

RUN;
%MEND;
%START;

PROC FREQ DATA=OUT1.DEMO; TABLES DEMOIND*CDRIND*SSIIND*SSDIIND/
LIST MISSING; TITLE 'CHECK MERGING'; RUN;

PROC CONTENTS DATA=OUT1.DEMO;
TITLE 'DEMO FOR TRF10';
RUN;

TITLE 'SSN LINKING';
PROC PRINT DATA=OUT2.LINKSSN (OBS=20); RUN;

PROC PRINT DATA=OUT1.DEMO (OBS=20);
```

Appendix A.53
JCL/SAS Code: T2T16DEM

```
TITLE 'AFTER UPDATE';
VAR  CAN SEX DOEI MINELGRD DOBNUM DOBREM DOBMBR DOBBEST DODBEST
FRSTNAME LASTNAME ENTDAT:
MIE_BEFORE_CDR DODEC: JUDLV: RID: RDT:
FRST_MIE;
FORMAT DODEC: DOBMBR DOBREM DOBBEST DODBEST YMMDD10.; RUN;

PROC PRINT DATA=OUT3.DMLDW(OBS=20);
TITLE 'DEDEDICATED LDW PROCESSING FILE';
VAR CAN DOBBEST DODBEST TWPCMLMNT1;
RUN;

TITLE 'DEMO FREQS';
PROC FREQ DATA=OUT1.DEMO;
TABLES RACE SEX SEXMISS CITIZEN RACE WHITE RACEMISS LANG
T16LANG
FRST_MIE MIE_BEFORE_CDR DUALELIG
DOBBEST DOBFLAG DOD DODBEST DODFLAG;
FORMAT DOBBEST DODBEST YMMDD10.; RUN;
```

Appendix A.54
SAS Code: Upload Provider Data

```
*****
* PROJECT: Evaluation of the Ticket to Work Program-Part A
*           Evaluation Implementation (6979) TRF10
* PROJECT
* DIRECTOR:   Craig Thornton
*
* PROGRAM:    M:\Page\TRF10\Programs\Upload Provider Data.sas
*
* DESCRIPTION: UPLOADS PROVIDER DATA
*
* DATE:       12/7/2011 Jeremy Page
***** ;

LIBNAME OUT 'M:\Page\TRF10\Data\PII';

PROC IMPORT OUT= OUT.EN_Provider_File_100411
            DATAFILE= 'M:\Page\TRF10\Data\PII\EN Provider File
100411.xls'
            DBMS=EXCEL REPLACE;
            SHEET='EN Provider File 100411';
            GETNAMES=YES;
            MIXED=YES;
            USEDATE=YES;
            SCANTIME=YES;
RUN;

PROC PRINT DATA = OUT.EN_Provider_File_100411;
RUN;

DATA OUT.TRUNC_PROVIDER
      (KEEP = Data_Universal_Numbering_System
Vocational_Rehabilitation_Switch);
      SET OUT.EN_Provider_File_100411;
RUN;

PROC PRINT DATA = OUT.TRUNC_PROVIDER;
RUN;

PROC SORT DATA = OUT.TRUNC_PROVIDER (RENAME =
(Data_Universal_Numbering_System = DUNS))
          OUT = OUT.PROVIDER
          NODUP;
          BY DUNS;
RUN;

SIGNON ;
libname local 'M:\Page\TRF10\Data\PII';
rsubmit;
```

Appendix A.54

SAS Code: Upload Provider Data

```
libname MISFDATA 'OPDR.TG.PRD.ETTW.#2127.TRF10P.PROVIDER.SA.V1' ;
option compress=binary;
*****;
/* Use the firstobs and obs control only if needed. For example, if
the first download */
/* after the 499th record, then the code below will start
downloading at the 500th */
/* observation and go until it completes or crashes.
*/
*****;

proc upload data=local.PROVIDER OUT=misfdata.PROVIDER
(compress=binary);

RUN;
endrsubmit;
signoff;
```

Appendix A.55
JCL/SAS Code: PROVFMF

```

//$2358PVF JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:   LAURA KOSARn
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(PROVFMF)
//*
//* DESCRIPTION: CREATE PROVIDER FORMAT FROM PROVIDER FILE
//*
//* DATE:       06/15/10 JEREMY PAGE
//* UPDATED:    12/20/11 NATALIE HAZELWOOD
//*****
*
//STEP1 EXEC SAS9
//WORK      DD SPACE=(6160,(99000,99000),,,ROUND)
//TEMP1 DD DSN=&&TEMP,DISP=(NEW,DELETE,DELETE),UNIT=TSILO,
//          VOL=(,,10)
//PROV                                           DD
DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.PROVIDER.SA.V1,DISP=SHR
//LIBRARY DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKT.FMTLIB,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//SYSIN DD *

* READ PROVIDER FILE;
DATA PROVIDER(KEEP=START LABEL FMTNAME);
  SET PROV.PROVIDER;

  * IN ORDER TO CREATE THE FORMAT THE VARIABLE CONTAINING THE VALUE
  * TO BE FORMATTED MUST BE NAMED "START" AND THE VARIABLE
CONTAINING
  * THE FORMATTED VALUE MUST BE NAME "LABEL";
  START=DUNS;
  LABEL=Vocational_Rehabilitation_Switch;
  ;

  IF LABEL='Y' THEN LABEL='V';
  ELSE IF LABEL='N' THEN LABEL='E';

  * A VARIABLE NAMED FMTNAME MUST BE CREATED TO STORE THE NAME OF
THE
  * FORMAT --- THE NAME USED TO REFERENCE THE FORMAT;
  FMTNAME='$PROVTYPE';

  OUTPUT;

  IF _N_=1 THEN DO;
    START='OTHER';
    LABEL=' ';
    OUTPUT;
  END;

```

Appendix A.55
JCL/SAS Code: PROVFMF

```
RUN;  
  
* CREATE THE FORMAT FROM THE FILE CREATED IN PREVIOUS DATASTEP;  
PROC FORMAT CNTLIN=PROVIDER LIBRARY=LIBRARY FMTLIB;  
RUN;  
  
* CHECK THAT THE FORMAT IS WORKING CORRECTLY BY PRINTING THE FORMATED  
  VALUE OF "START" FROM THE CNTLIN FILE AND THE VALUE OF "LABEL". IF  
  THE FORMAT CREATION WAS SUCCESSFULL THESE VALUES WILL BE THE SAME;  
PROC PRINT DATA=PROVIDER;  
  VAR START LABEL;  
  FORMAT START $PROVTYPE. ;  
RUN;
```

Appendix A.56
JCL/SAS Code: DCFEXT

```
// $2358DCF JOB (12510000,T715,,SAS,,ITC9TH),2358HAZE,CLASS=T,
// MSGCLASS=1,MSGLEVEL=(1,1),REGION=4096K,NOTIFY=$2358
/*SSAMAIN SASBASIC
//*****
/* PROJECT: Evaluation of the Ticket to Work Program-Part A
/*           Evaluation Implementation (06979) TRF10
/* PROJECT
/* DIRECTOR:   LAURA KOSAR
/*
/* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.DCF.PRDLIB(DCFEXT)
/*
/* DESCRIPTION: CREATE DCF EXTRACTS FOR TRF TICKET FILE
/*
/* DATE:       06/16/10 DAWN PHELPS
/* UPDATED FOR TRF10: 10/24/11 NATALIE HAZELWOOD
//*****
//S1 EXEC SAS9,LOAD='DBP8.DB2.SDSNLOAD',SORT='15000',
//      WORK='15000,15000',COND=EVEN
//TKTASGN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCF TKTA.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=TSILO,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HTKTASGN DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHTKTA.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//TKT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCF TKT.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HTKT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHTKT.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//TKTTITLE DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCF TKTT.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HTKTTITL DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHTKTT.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//TKTMNTH  DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCF TKTM.SA.V1,
//  DISP=(,CATLG,DELETE),
//  UNIT=AFF=TKTASGN,
//  VOL=(,,255),
//  DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
```

Appendix A.56
JCL/SAS Code: DCFEXT

```
//HTKTMNTH DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHTKTM.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=TKTASGN,
// VOL=(,,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//CLNT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFCLNT.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=TKTASGN,
// VOL=(,,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//SYSIN DD *
```

options nocenter;

```
LIBNAME TKTASGN TAPE;
LIBNAME HTKTASGN TAPE;
LIBNAME TKT TAPE;
LIBNAME HTKT TAPE;
LIBNAME TKTTITLE TAPE;
LIBNAME HTKTTITL TAPE;
LIBNAME TKTMNTH TAPE;
LIBNAME HTKTMNTH TAPE;
LIBNAME CLNT TAPE;
LIBNAME MDCF DB2 SSID=DBP8 SCHEMA=MDCF ;
```

```
PROC SQL;
CREATE TABLE TKTASGN.TKTASGN AS
SELECT * FROM MDCF.TKTASGN;
QUIT;
PROC SQL;
CREATE TABLE HTKTASGN.HTKTASGN AS
SELECT * FROM MDCF.HTKTASGN;
QUIT;
PROC SQL;
CREATE TABLE TKT.TKT AS
SELECT * FROM MDCF.TKT;
QUIT;
PROC SQL;
CREATE TABLE HTKT.HTKT AS
SELECT * FROM MDCF.HTKT;
QUIT;
PROC SQL;
CREATE TABLE TKTTITLE.TKTTITLE AS
SELECT * FROM MDCF.TKTTITLE;
QUIT;
PROC SQL;
CREATE TABLE HTKTTITL.HTKTTITL AS
SELECT * FROM MDCF.HTKTTITL;
QUIT;
PROC SQL;
CREATE TABLE TKTMNTH.TKTMNTH AS
SELECT * FROM MDCF.TKTMNTH;
QUIT;
```

Appendix A.56
JCL/SAS Code: DCFEXT

```
PROC SQL;
CREATE TABLE HTKTMNTH.HTKTMNTH AS
SELECT * FROM MDCF.HTKTMNTH;
QUIT;
PROC SQL;
CREATE TABLE CLNT.CLNT          AS
SELECT * FROM MDCF.CLNT;
QUIT;

RUN;
PROC CONTENTS DATA=TKTASGN.TKTASGN;
PROC CONTENTS DATA=HTKTASGN.HTKTASGN;
PROC CONTENTS DATA=TKT.TKT;
PROC CONTENTS DATA=HTKT.HTKT;
PROC CONTENTS DATA=TKTTITLE.TKTTITLE;
PROC CONTENTS DATA=HTKTTITL.HTKTTITL;
PROC CONTENTS DATA=TKTMNTH.TKTMNTH;
PROC CONTENTS DATA=HTKTMNTH.HTKTMNTH;
PROC CONTENTS DATA=CLNT.CLNT;
RUN;
```

Appendix A.57
JCL/SAS Code: TKTTITLE

```

//$2358TLE  JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1)
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:    LAURA KOSAR
//*
//* PROGRAM:     OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(TKTTITLE)
//*
//* DESCRIPTION: CREATE TICKET TITLE & IN-USE DATA
//*
//* DATE:        06/22/10 DAWN PHELPS (FERRAGAMO)
//* UPDATED:     12/20/11 NATALIE HAZELWOOD
//*****
//SAS        EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1  DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCF TKM.SA.V1,DISP=SHR
//OUT  DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKTTITLE.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//TEMP1 DD   DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD  *

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX;

PROC SORT DATA=IN1.TKTMNTH OUT=TEMP1.TKTMNTH;
  WHERE SSACT_PRTCPN_CD NE '0';
  BY COSSN TKT_STMDT;
RUN;

DATA TEMP1.TKTMNTH(KEEP=COSSN TKT_STMDT SSACT_PRTCPN_CD Yymm
  TKT_INUSE_SW);
  SET TEMP1.TKTMNTH;
  BY COSSN TKT_STMDT;

  * CREATE VARIABLE CONTAINING THE Yymm OF THE EFFECTIVE DATE;
  LENGTH Yymm $4.;
  Yymm=PUT(TKT_STMDT,YymmDD4.);

  * KEEP ALL UNIQUE COMBINATIONS OF COSSN AND TKT_STMDT;
  * IF NOT A UNIQUE COMBO THEN KEEP THE FIRST OBS -- TESTS SHOW
THAT
  THIS HAPPENS FOR ONLY 12 CASES;
  IF FIRST.TKT_STMDT THEN OUTPUT;

RUN;

PROC SORT DATA=TEMP1.TKTMNTH;
  BY COSSN Yymm;
RUN;

```

Appendix A.57
JCL/SAS Code: TKTTITLE

```
* CREATE ONE RECORD PER COSSN WITH MONTHLY TITLE VARIABLES;
PROC TRANSPOSE DATA=TEMP1.TKTMNTH OUT=TEMP1.TKTTITLE PREFIX=TITLE_MO;
  BY COSSN;
  ID YMM;
  VAR SSACT_PRTCPN_CD;
RUN;

* CREATE ONE RECORD PER COSSN WITH MONTHLY INUSE VARIABLES;
PROC          TRANSPOSE          DATA=TEMP1.TKTMNTH          OUT=TEMP1.TKTINUSE
PREFIX=CINUSE_MO;
  BY COSSN;
  ID YMM;
  VAR TKT_INUSE_SW;
RUN;

* POPULATE ALL MONTHLY TITLE & IN-USE VARIABLES;
DATA OUT.TKTTITLE;
  MERGE TEMP1.TKTTITLE(IN=A)
        TEMP1.TKTINUSE(IN=B)
        ;
  BY COSSN;

  ARRAY TITLE_MO $ TITLE_MO0201-TITLE_MO0212
                  TITLE_MO0301-TITLE_MO0312
                  TITLE_MO0401-TITLE_MO0412
                  TITLE_MO0501-TITLE_MO0512
                  TITLE_MO0601-TITLE_MO0612
                  TITLE_MO0701-TITLE_MO0712
                  TITLE_MO0801-TITLE_MO0812
                  TITLE_MO0901-TITLE_MO0912
                  TITLE_MO1001-TITLE_MO1012
                  ;

  ARRAY CINUSE_MO (*) CINUSE_MO0201-CINUSE_MO0212
                    CINUSE_MO0301-CINUSE_MO0312
                    CINUSE_MO0401-CINUSE_MO0412
                    CINUSE_MO0501-CINUSE_MO0512
                    CINUSE_MO0601-CINUSE_MO0612
                    CINUSE_MO0701-CINUSE_MO0712
                    CINUSE_MO0801-CINUSE_MO0812
                    CINUSE_MO0901-CINUSE_MO0912
                    CINUSE_MO1001-CINUSE_MO1012
                    ;

  ARRAY INUSE_MO (*) INUSE_MO0201-INUSE_MO0212
                   INUSE_MO0301-INUSE_MO0312
                   INUSE_MO0401-INUSE_MO0412
                   INUSE_MO0501-INUSE_MO0512
                   INUSE_MO0601-INUSE_MO0612
                   INUSE_MO0701-INUSE_MO0712
                   INUSE_MO0801-INUSE_MO0812
                   INUSE_MO0901-INUSE_MO0912
                   INUSE_MO1001-INUSE_MO1012
                   ;
```

Appendix A.57
JCL/SAS Code: TKTTITLE

```
* IF THE CURRENT MONTH IS NOT BLANK AND THE SUBSEQUENT
  MONTH IS BLANK THEN POPULATE THE SUBSEQUENT MONTH
  WITH THE CURRENT MONTH.  THIS WILL POPULATE ALL MONTHS
  FROM THE FIRST MONTH OF EFFECTIVE DATE UNTIL THE
  MONTH THE VARIABLE CHANGES.;
DO I=1 TO DIM(TITLE_MO)-1;

  IF TITLE_MO(I) NE '' AND TITLE_MO(I+1)=''
  THEN TITLE_MO(I+1)=TITLE_MO(I);

  IF CINUSE_MO(I) NE '' AND CINUSE_MO(I+1)=''
  THEN CINUSE_MO(I+1)=CINUSE_MO(I);

END;

* SET NUMERIC INUSE FLAG;
DO I=1 TO DIM(INUSE_MO);
  IF CINUSE_MO(I)='Y' THEN INUSE_MO(I)=1;
  ELSE IF CINUSE_MO(I)='N' THEN INUSE_MO(I)=0;
  ELSE INUSE_MO(I)=.;
END;

RUN;

PROC CONTENTS DATA=OUT.TKTTITLE;
RUN;

PROC PRINT DATA=OUT.TKTTITLE(OBS=100);
  VAR COSSN TITLE_MO0201-TITLE_MO0212
            TITLE_MO0301-TITLE_MO0312
            TITLE_MO0401-TITLE_MO0412
            TITLE_MO0501-TITLE_MO0512
            TITLE_MO0601-TITLE_MO0612
            TITLE_MO0701-TITLE_MO0712
            TITLE_MO0801-TITLE_MO0812
            TITLE_MO0901-TITLE_MO0912
            TITLE_MO1001-TITLE_MO1012
            INUSE_MO0201-INUSE_MO0212
            INUSE_MO0301-INUSE_MO0312
            INUSE_MO0401-INUSE_MO0412
            INUSE_MO0501-INUSE_MO0512
            INUSE_MO0601-INUSE_MO0612
            INUSE_MO0701-INUSE_MO0712
            INUSE_MO0801-INUSE_MO0812
            INUSE_MO0901-INUSE_MO0912
            INUSE_MO1001-INUSE_MO1012
            ;
RUN;

ENDSAS;
```

Appendix A.58
JCL/SAS Code: TKTMLASG

```

//$2358TKT JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(TKTMLASG)
//*
//* DESCRIPTION: CREATE TICKET MAILINGS & ASSIGNMENT PORTIONS OF
TRF10
//*
//* DATE:       06/18/10 DAWN PHELPS (FERRAGAMO)
//* UPDATED:    12/20/11 NATALIE HAZELWOOD FOR TRF10
//* PURPOSE:    FLATTEN TICKET MAILED & ASSIGNMENT FILES TO ONE PER SSN
//*****
//*
//TKTMAIL    EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFTKT.SA.V1,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFTKTA.SA.V1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKTMLASG.SA.V1,
//      DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//      UNIT=TSILO
//LIBRARY DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKT.FMTLIB,DISP=SHR
//TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX;

PROC SORT DATA=IN1.TKT OUT=TEMP1.TKT;
  BY COSSN TKT_NUM;
RUN;

PROC SORT DATA=IN2.TKTASGN OUT=TEMP2.TKTASGN;
  BY COSSN TKT_NUM;
RUN;

DATA      OUT.TKTMLASG(KEEP=TKTMAILDDT:      TKTTERMDDT:      TKTASGNDDT:
TKTUNASGDT:
          DUNS: PROVTYPE: PMTTYPER: NOE COSSN NUMBER:);

  RETAIN TKTMAILDDT1-TKTMAILDDT30
         TKTTERMDDT1-TKTTERMDDT30
         TKTASGNDDT1-TKTASGNDDT30
         TKTUNASGDT1-TKTUNASGDT30
         DUNS1-DUNS30
         PROVTYPE1-PROVTYPE30
         PMTTYPER1-PMTTYPER30

```

Appendix A.58
JCL/SAS Code: TKTMLASG

```
NOE
NUMBER1-NUMBER30
;

MERGE TEMP1.TKT(IN=MAIL)
      TEMP2.TKTASGN(IN=ASGN)
;
BY COSSN TKT_NUM;

* OUTPUT ARRAYS;
ARRAY TKTMAILDDT TKTMAILDDT1-TKTMAILDDT30;
ARRAY TKTTERMDDT TKTTERMDDT1-TKTTERMDDT30;
ARRAY TKTASGNDDT TKTASGNDDT1-TKTASGNDDT30;
ARRAY TKTUNASGDT TKTUNASGDT1-TKTUNASGDT30;
ARRAY DUNSA $9. DUNSA1-DUNSA30;
ARRAY PMTTYPER $ PMTTYPER1-PMTTYPER30;
ARRAY PROVTYPE $ PROVTYPE1-PROVTYPE30;
ARRAY NUMBER NUMBER1-NUMBER30;

* INITIALIZE VARIABLES WHEN ENCOUNTERING A NEW SSN;
IF FIRST.COSSN THEN DO;
  DO I=1 TO 30;
    TKTMAILDDT(I)=.;
    TKTTERMDDT(I)=.;
    TKTASGNDDT(I)=.;
    TKTUNASGDT(I)=.;
    DUNSA(I)=' ';
    PMTTYPER(I)=' ';
    PROVTYPE(I)=' ';
    NUMBER(I)=.;
  END;
  NOE=0;
END;

* SET NUMBER OF ENTRIES (NOE) VARIABLE;
NOE=NOE+1;

* POPULATE THE TICKET ACTIVE VARIABLES;
TKTMAILDDT(NOE)=MAILD_DT;
TKTASGNDDT(NOE)=ASGND_DT;
TKTUNASGDT(NOE)=UNASGND_DT;
DUNSA(NOE)=DUNSA;
PMTTYPER(NOE)=TKT_PMT_TYP;
PROVTYPE(NOE)=PUT(DUNSA(NOE), $PROVTYPE.);
IF TKT_STUS_CD='T' THEN TKTTERMDDT(NOE)=TKT_STUS_DT;
NUMBER(NOE)=TKT_NUM;

* IF THE LAST RECORD FOR THIS SSN THEN OUTPUT;
IF LAST.COSSN THEN OUTPUT;

RUN;

PROC CONTENTS DATA=OUT.TKTMLASG;
```

Appendix A.58
JCL/SAS Code: TKTMLASG

RUN;

TITLE1 'SAMPLE PRINT:';

TITLE2 'OPDR.TG.PRD.ETTW.\$2358.TRF10P.TKTMLASG.SA.V1(TKTMLASG)';

TITLE3 'FROM OPDR.TG.PRD.ETTW.\$2358.TRF10.TKT.PRDLIB(TKTMLASG)';

PROC PRINT DATA=OUT.TKTMLASG(OBS=50);

FORMAT TKTMAILDDT: TKTASGNDDT: TKTUNASGDT: TKTTERMDDT: YMMDD10.;

RUN;

ENDSAS;

Appendix A.59
JCL/SAS Code: BLDPRETK

```
// $2358BLD JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//*
//* PROJECT DIRECTOR: LAURA KOSAR
//*
//* PROGRAM: OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(BLDPRETK)
//*
//* DATE:          06/24/10 DAWN PHELPS
//* UPDATED:      12/20/11 NATALIE HAZELWOOD
//*
//* PURPOSE: COMBINE NEW DCF EXTRACT TABLES TO CREATE A FILE THAT
//*           MIMICS THE STRUTURE OF THE TRF05 INPUT FILE
//*****
//*
//TKTMAIL   EXEC SAS9,
//          WORK='120000,60000'
//*
//TKT  DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKTMLASG.SA.V1,DISP=SHR
//TLE  DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKTTITLE.SA.V1,DISP=SHR
//CLNT DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFCLNT.SA.V1,
//          DISP=SHR
//*
//OUT  DD   DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.BLDPRETK.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//LIBRARY DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKT.FMTLIB,DISP=SHR
//TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *
```

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX;

* COMBINE TICKET MAIL & ASSIGNMENT FILE WITH TICKET TITLE FILE;

```
DATA TEMP1.TKT;
  MERGE TKT.TKTMLASG(IN=TKT)
        TLE.TKTTITLE(IN=TLE)
        ;
  BY COSSN;
  IF TKT;
```

RUN;

* SORT DCF CLIENT DATASET AND KEEP ONLY WHERE THE COSSN IS THE BENEFICIARIES BOAN;

```
PROC SORT DATA=CLNT.CLNT OUT=TEMP2.CLNT NODUPKEYS;
  WHERE CID='00'; /* COSSN IS THE BENEFICIARIES BOAN */
  BY COSSN;
```

RUN;

Appendix A.59
JCL/SAS Code: BLDPRETK

```

* COMBINE TICKET MAIL, ASSIGNMENT, & TITLE FILE WITH CLNT FILE TO
  OBTAIN THE DOB & DOD;
DATA OUT.BLDPRETK(DROP=T16_DOB T2_DOB T16_DOD T2_DOD);
  MERGE TEMP1.TKT(IN=TKT)
        TEMP2.CLNT(IN=CLNT KEEP=T16_DOB T2_DOB T16_DOD T2_DOD COSSN
                  )
        ;
  BY COSSN;
  IF TKT;

  IF CLNT THEN INCLNT=1;
  ELSE INCLNT=0;

* IN TRF07 WE HAD TO SWITCH THE SOURCE OF DOB & DOD.  PREVIOUSLY
  THESE VARIABLES CAME FROM THE DCF GROUP 1 PERSON FILE.  NOW WE
  ARE USING THE DOBBEST & DODBEST FIELDS FROM THE TRF08 DEMO
FILE;

* IN TRF10 WE SWITCHED TO USING THE DCF CLIENT TABLE FOR DOB
  AND DOD RATHER THAN USING THE DEMO FILE;

* DETERMINE DOBBEST;
DOBBEST_TKT=T16_DOB;
IF DOBBEST_TKT=. THEN DOBBEST_TKT=T2_DOB;

* DETERMINE DODBEST;
DODBEST_TKT=T16_DOD;
IF DODBEST_TKT=. THEN DODBEST_TKT=T2_DOD;

*****          CALCULATE          FULL          RETIREMENT          AGE
*****          ;
* PREVIOUS VERISONS OF THE TRF ONLY CONSIDERD A PERSON TICKET
* ELIGIBLE IF HE WAS BETWEEN THE AGES OF 18 & 65.  WITH TRF08 IT
* HAS BEEN DECIDED THAT WE WILL CONSIDER A PERSON TICKET ELIGIBLE
* IF HE IS BETWEEN THE AGES OF 18 & FRA.  TO THAT END FRA IS BEING
* CALUCLATED HERE TO BE USED IN THE CRTKTFLG PROGRAM. ;

* ADJUST RETIREMENT DATE FOR BENEFICIARIES BORN ON JAN 1 ;
IF MONTH(DOBBEST_TKT)=1 AND DAY(DOBBEST_TKT)=1 THEN
  DOBRETIRE=INTNX('DAY',DOBBEST_TKT,-1);
ELSE DOBRETIRE=DOBBEST_TKT;

* SET FRA;
IF YEAR(DOBRETIRE)<=1937 THEN FRA=780/12;          /* 65 */
ELSE IF YEAR(DOBRETIRE)=1938 THEN FRA=782/12;          /* 65 & 2 M
*/
ELSE IF YEAR(DOBRETIRE)=1939 THEN FRA=784/12;          /* 65 & 4 M
*/
ELSE IF YEAR(DOBRETIRE)=1940 THEN FRA=786/12;          /* 65 & 6 M
*/
ELSE IF YEAR(DOBRETIRE)=1941 THEN FRA=788/12;          /* 65 & 8 M
*/
ELSE IF YEAR(DOBRETIRE)=1942 THEN FRA=790/12;          /* 65 & 10 M
*/

```

Appendix A.59
JCL/SAS Code: BLDPRETK

```
ELSE IF 1943<=YEAR(DOBRETIRE)<=1954 THEN FRA=792/12; /* 66 */
ELSE IF YEAR(DOBRETIRE)=1955 THEN FRA=794/12; /* 66 & 2 M
*/
ELSE IF YEAR(DOBRETIRE)=1956 THEN FRA=796/12; /* 66 & 4 M
*/
ELSE IF YEAR(DOBRETIRE)=1957 THEN FRA=798/12; /* 66 & 6 M
*/
ELSE IF YEAR(DOBRETIRE)=1958 THEN FRA=800/12; /* 66 & 8 M
*/
ELSE IF YEAR(DOBRETIRE)=1959 THEN FRA=802/12; /* 66 & 10 M
*/
ELSE IF YEAR(DOBRETIRE)>=1960 THEN FRA=804/12; /* 67 */

RUN;

TITLE1;
TITLE2;
PROC CONTENTS DATA=OUT.BLDPRETK;
RUN;

TITLE1 'SAMPLE PRINT: ';
TITLE2 'OPDR.TG.PRD.ETTW.$2358.TRF10P.BLDPRETK.SA.V1';
TITLE3 'FROM OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(BLDPRETK)';
PROC PRINT DATA=OUT.BLDPRETK(OBS=50);
FORMAT TKTMAILDDT: TKTASGNDDT: TKTUNASGDT: TKTTERMDDT:
DOBBEST_TKT
DODBEST_TKT YYMMDD10.;
RUN;

ENDSAS;
```

Appendix A.60
JCL/SAS Code: ADRD1VAR

```

//$2358TKT JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:    LAURA KOSAR
//*
//* PROGRAM:     OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(ADRD1VAR)
//*
//* DESCRIPTION: ADD BENES FROM PREVIOUS ROUNDS
//*
//* DATE:        06/25/10 DAWN PHELPS
//* UPDATED:     12/20/11 NATALIE HAZELWOOD
//*****
*
//STEP1 EXEC SAS9
//WORK      DD SPACE=(6160,(99000,99000),,,ROUND)
//INTRF10                                       DD
DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.BLDPRETK.SA.V1,DISP=SHR
//INTRF09                                       DD
DSN=OPDR.TG.PRD.ETTW.N4671.TRF09P.CRTKTFLG.SA.V1,DISP=SHR
//LIBRARY  DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TKT.FMTLIB,DISP=SHR
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ADRD1VAR.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//SYSIN    DD *

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX;

%MACRO START;
DATA OUT.ADRD1VAR;
    MERGE INTRF10.BLDPRETK(IN=INTRF10
                          RENAME=(DOBBEST_TKT=DOBTKT
                                   DODBEST_TKT=DODTKT))
          INTRF09.TKTTFM(IN=INTRF09      KEEP=COSSN      TRF06_PHASE
TITLE_MO0912
                          DOBTKT TKTEIN1-TKTEIN5
                          TRF06_TKTSLTDDT
                          TKTMAILDDT1-TKTMAILDDT5
                          TKTASGNDDT1-TKTASGNDDT5
                          TKTUNASGDT1-TKTUNASGDT5
                          TKTTTERMDDT1-TKTTTERMDDT5
                          RD1_: RD2_:
                          TRF1 TRF2 TRF05 TRF06 TRF07 TRF08
                          TRF09
                          RENAME=(TITLE_MO0912=TRF09_TITLE_MO0912
                                   DOBTKT=TRF09_TKTDOB
                                   %DO I=1 %TO 5;
                                   TKTEIN&I.=TRF09_TKTEIN&I.
                                   TKTMAILDDT&I.=TRF09_TKTMAILDDT&I.
                                   TKTASGNDDT&I.=TRF09_TKTASGNDDT&I.
                                   TKTUNASGDT&I.=TRF09_TKTUNASGDT&I.
                                   TKTTTERMDDT&I.=TRF09_TKTTTERMDDT&I.

```

Appendix A.60
JCL/SAS Code: ADRD1VAR

```
                                %END;
                                ))

                                ;
                                BY COSSN;

                                * CREATE FLAGS FOR WHICH ROUND A BENE WAS FOUND IN;
                                IF INTRF10 THEN TRF10=1;
                                ELSE DO;

                                    TRF10=0;

                                    * MOVE PREVIOUS DATA TO CURRENT VARIABLES;
                                    *LENGTH PROVTYPE1-PROVTYPE5 $1.;
                                    ARRAY PROVTYPE PROVTYPE1-PROVTYPE5;
                                    ARRAY TKTEIN $ TKTEIN1-TKTEIN5;
                                    ARRAY MAILDT TKTMAILDDT1-TKTMAILDDT5;
                                    ARRAY PREV_MAILDT TRF09_TKTMAILDDT1-TRF09_TKTMAILDDT5;
                                    ARRAY TERMDT TKTTERMDDT1-TKTTERMDDT5;
                                    ARRAY PREV_TERMDT TRF09_TKTTERMDDT1-TRF09_TKTTERMDDT5;
                                    ARRAY ASGNDT TKTASGNDDT1-TKTASGNDDT5;
                                    ARRAY PREV_ASGNDT TRF09_TKTASGNDDT1-TRF09_TKTASGNDDT5;
                                    ARRAY UNASSGDT TKTUNASGDT1-TKTUNASGDT5;
                                    ARRAY PREV_UNASSGDT TRF09_TKTUNASGDT1-TRF09_TKTUNASGDT5;

                                DO I=1 TO 5;
                                    PROVTYPE(I)=PUT(TKTEIN(I), $PROVTYPE.);
                                    MAILDT(I)=PREV_MAILDT(I);
                                    TERMDT(I)=PREV_TERMDT(I);
                                    ASGNDT(I)=PREV_ASGNDT(I);
                                    UNASSGDT(I)=PREV_UNASSGDT(I);
                                END;

                                TKTDOB=TRF09_TKTDOB;

                                NOE=5;

                                END;

                                * SET FILE CONTRIBUTION FLAG TO ZERO IF MISSING;
                                ARRAY CONFILE (*) TRF1 TRF2 TRF05 TRF06 TRF07 TRF08 TRF09 TRF10;
                                DO I=1 TO DIM(CONFILE);
                                    IF CONFILE(I)=. THEN CONFILE(I)=0;
                                END;

                                RUN;

                                PROC CONTENTS;
                                RUN;

                                PROC FREQ DATA=OUT.ADRD1VAR;
                                    TABLE TRF1;
                                    TABLE TRF2;
                                    TABLE TRF05;
```

Appendix A.60
JCL/SAS Code: ADRD1VAR

```
TABLE TRF06;
TABLE TRF08;
TABLE TRF07;
TABLE TRF09;
TABLE TRF10;
RUN;

TITLE1 'TKT MAIL DATE DISTRIBUTION FOR BENES IN TRF10 ONLY';
PROC FREQ DATA=OUT.ADRD1VAR;
  WHERE TRF10=1 AND TRF1=0 AND TRF2=0 AND TRF05=0 AND TRF06=0
        AND TRF07=0 AND TRF08=0 AND TRF09=0;
  FORMAT TKTMAILDDT1-TKTMAILDDT5 YMMDD10.;
  TABLE TKTMAILDDT1;
  TABLE TKTMAILDDT2;
  TABLE TKTMAILDDT3;
  TABLE TKTMAILDDT4;
  TABLE TKTMAILDDT5;
RUN;

*** THIS CODE HAS BEEN REMOVED AND PUT INTO ANOTHER PROGRAM ***;
*** IN ORDER TO AVOID ANY ISSUES OF ACCIDENTALLY HAVING PII ***;
*** RELEASED ***;
*PROC PRINT DATA=OUT.ADRD1VAR(OBS=100);
*RUN;

%MEND START;
%START;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
// $2358CTK JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
// *****
// * PROJECT: BUILD TRF10 (06979)
// *
// * PROJECT
// * DIRECTOR:    LAURA KOSAR
// *
// * PROGRAM:     OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(CRTKTFLG)
// *
// * DESCRIPTION: CREATE TICKET PORTION OF TRF
// *
// * DATE:        07/07/10 DAWN PHELPS (FERRAGAMO)
// * UPDATED:     1/26/12 NATALIE HAZELWOOD
// * PURPOSE:     CREATE TICKET PORTION OF THE TICKET RESEARCH FILE.
// * PROGRAM CREATES MONTHLY FLAGS (ELIGIBLE, ACTIVE, PMTTYPE,
// * PROVTYPE, ETC. - SEE SPECS "VARIABLES FOR TICKET MASTER
// * FILE, TICKET PORTION ONLY") BASED ON DCF TKT DATA (TKTNUM,
// * TKTMAILDDT, TKTASGNDDT, TKTUNASGDT, TKTTERMDDT, ETC.).
// *****
// *
// SAS          EXEC SAS9,
//              WORK='120000,60000'
// *
// IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ADRD1VAR.SA.V1,DISP=SHR
// OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.CRTKTFLG.SA.V1,
//       DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//       UNIT=TSILO
// TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//        SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
// SYSIN DD *

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX;

TITLE1 "BUILD TRF10";
TITLE2 "TKT FLAG CREATION";
TITLE3;

/* change endyr and endmmn for each run */
%let begyr=2002;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */
%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
    %if &j<10 %then %let mn=0%eval(&j);
        %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
%end;
%end;
%let tot=%eval(&k-1);

DATA OUT.TKTTMF;
  SET IN1.ADRD1VAR;

  ARRAY ELIGIBLE_MO (*) %DO I=1 %TO &TOT; ELIGIBLE_MO&&X&I %END;;
  ARRAY ACTIVE_MO (*) %DO I=1 %TO &TOT; ACTIVE_MO&&X&I %END;;
  ARRAY PMTTYE_MO (*) $ %DO I=1 %TO &TOT; PMTTYE_MO&&X&I %END;;
  ARRAY PROVTYPE_MO (*) $ %DO I=1 %TO &TOT; PROVTYPE_MO&&X&I %END;;
  ARRAY ELIGIBLE_EV (*) %DO I=1 %TO &TOT; ELIGIBLE_EV&&X&I %END;;
  ARRAY ACTIVE_EV (*) %DO I=1 %TO &TOT; ACTIVE_EV&&X&I %END;;
  ARRAY VR_EV (*) %DO I=1 %TO &TOT; VR_EV&&X&I %END;;
  ARRAY EN_EV (*) %DO I=1 %TO &TOT; EN_EV&&X&I %END;;
  ARRAY MAIL_MO (*) %DO I=1 %TO &TOT; MAIL_MO&&X&I %END;;
  ARRAY FIRST_ASGN_MO (*) %DO I=1 %TO &TOT; FIRST_ASGN_MO&&X&I %END;;
  ARRAY DEACTIVE_MO (*) %DO I=1 %TO &TOT; DEACTIVE_MO&&X&I %END;;
  ARRAY REASSIGN_MO (*) %DO I=1 %TO &TOT; REASSIGN_MO&&X&I %END;;
  ARRAY FIRST_ASGN_VR_MO (*) %DO I=1 %TO &TOT;
    FIRST_ASGN_VR_MO&&X&I %END;;
  ARRAY DEACTIVE_VR_MO (*) %DO I=1 %TO &TOT;
    DEACTIVE_VR_MO&&X&I %END;;
  ARRAY REASSIGN_VR_MO (*) %DO I=1 %TO &TOT;
    REASSIGN_VR_MO&&X&I %END;;
  ARRAY FIRST_ASGN_EN_MO (*) %DO I=1 %TO &TOT;
    FIRST_ASGN_EN_MO&&X&I %END;;
  ARRAY DEACTIVE_EN_MO (*) %DO I=1 %TO &TOT;
    DEACTIVE_EN_MO&&X&I %END;;
  ARRAY REASSIGN_EN_MO (*) %DO I=1 %TO &TOT;
    REASSIGN_EN_MO&&X&I %END;;

  ARRAY VR_TR_MO (*) %DO I=1 %TO &TOT; VR_TR_MO&&X&I %END;;
  ARRAY VR_MO_MO (*) %DO I=1 %TO &TOT; VR_MO_MO&&X&I %END;;
  ARRAY VR_OO_MO (*) %DO I=1 %TO &TOT; VR_OO_MO&&X&I %END;;
  ARRAY EN_MO_MO (*) %DO I=1 %TO &TOT; EN_MO_MO&&X&I %END;;
  ARRAY EN_OO_MO (*) %DO I=1 %TO &TOT; EN_OO_MO&&X&I %END;;
  ARRAY VR_TR_EV (*) %DO I=1 %TO &TOT; VR_TR_EV&&X&I %END;;
  ARRAY VR_MO_EV (*) %DO I=1 %TO &TOT; VR_MO_EV&&X&I %END;;
  ARRAY VR_OO_EV (*) %DO I=1 %TO &TOT; VR_OO_EV&&X&I %END;;
  ARRAY EN_MO_EV (*) %DO I=1 %TO &TOT; EN_MO_EV&&X&I %END;;
  ARRAY EN_OO_EV (*) %DO I=1 %TO &TOT; EN_OO_EV&&X&I %END;;

  ARRAY TKTNUM NUMBER1-NUMBER30;
  ARRAY MAILDT TKTMAILDDT1-TKTMAILDDT30;
  ARRAY TERMDT TKTTERMDDT1-TKTTERMDDT30;
  ARRAY ASGNDT TKTASGNDDT1-TKTASGNDDT30;
  ARRAY UNASSGDT TKTUNASGDT1-TKTUNASGDT30;
  ARRAY PMTTYE PMTTYE1-PMTTYE30;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
ARRAY PROVTYPE PROVTYPE1-PROVTYPE30;

* SET BEGINNING AND ENDING DATES FOR EACH REPORTING MONTH;
* I.E. BEGDATE1=20020101 ENDDATE1=20020131;

FORMAT BEGDATE: ENDDATE: MMDDYY10.;

ARRAY BEGDATE (*) BEGDATE1-BEGDATE&TOT.;
ARRAY ENDDATE (*) ENDDATE1-ENDDATE&TOT.;

DO YR=&BEGYR. TO &ENDYR.;
  IF YR NE &ENDYR. THEN DO MO=1 TO 12;
    BEGDATE(((YR-&BEGYR.)*12)+MO)=MDY(MO,1,YR);
    ENDDATE(((YR-&BEGYR.)*12)+MO)
      =INTNX('MONTH',BEGDATE(((YR-&BEGYR.)*12)+MO),0,'END');
  END;
  ELSE DO MO=1 TO &ENDMN.;
    BEGDATE(((YR-&BEGYR.)*12)+MO)=MDY(MO,1,YR);
    ENDDATE(((YR-&BEGYR.)*12)+MO)
      =INTNX('MONTH',BEGDATE(((YR-&BEGYR.)*12)+MO),0,'END');
  END;
END;

* SET TITLE VARIABLE FOR BENES APPEARING ONLY IN PREVIOUS VERSIONS;
IF TRF10 NE 1 THEN TITLE_MO1012=TRF09_TITLE_MO0912;

* INITIALIZE EVER VARIABLES;
EVER_ELIG=0;
EVER_ACTIVE=0;
EVER_VR=0;
EVER_EN=0;
EVER_VR_TR=0;
EVER_VR_MO=0;
EVER_VR_OO=0;
EVER_EN_MO=0;
EVER_EN_OO=0;

*** LOOP THROUGH ONCE FOR EACH REPORTING MONTH;
DO RPTMO=1 TO DIM(BEGDATE);

  *** CALCULATE AGE FOR REPORTING MONTH;
  * OLD CALCULATION: AGE=YRDIF(DOBTKT,ENDDATE(RPTMO),'ACT/ACT');
  AGE=INTCK('MONTH',DOBTKT,ENDDATE(RPTMO))/12;

  *** INITIALIZE VARIABLES;
  ELIGIBLE_MO(RPTMO)=0;
  ACTIVE_MO(RPTMO)=0;
  PMTTYPE_MO(RPTMO)=' ';
  PROVTYPE_MO(RPTMO)=' ';
  ELIGIBLE_EV(RPTMO)=0;
  ACTIVE_EV(RPTMO)=0;
  MAIL_MO(RPTMO)=0;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
VR_EV(RPTMO)=0;
EN_EV(RPTMO)=0;
VR_TR_MO(RPTMO)=0;
VR_MO_MO(RPTMO)=0;
VR_OO_MO(RPTMO)=0;
EN_MO_MO(RPTMO)=0;
EN_OO_MO(RPTMO)=0;
FIRST_ASGN_MO(RPTMO)=0;
FIRST_ASGN_VR_MO(RPTMO)=0;
FIRST_ASGN_EN_MO(RPTMO)=0;
DEACTIVE_MO(RPTMO)=0;
DEACTIVE_VR_MO(RPTMO)=0;
DEACTIVE_EN_MO(RPTMO)=0;
REASSIGN_MO(RPTMO)=0;
REASSIGN_VR_MO(RPTMO)=0;
REASSIGN_EN_MO(RPTMO)=0;

*** FLAG TICKET MAILED IN REPORTING MONTH;

*** TICKET HAS BEEN MAILED IF ALL OF THE FOLLOWING ARE TRUE:
1. THE MAIL DATE IS ON OR AFTER THE FIRST DAY OF THE
REPORTING MONTH OR ON OR BEFORE THE LAST DAY OF THE REPORTING MONTH
2. THE TICKET TERMINATION DATE IS MISSING OR IS AFTER THE
MAIL DATE
4. THE MAIL DATE IS NOT MISSING
;
TKTEVT=1;
*** LOOP THRU TICKETS UNTIL MAILED TICKET IS FOUND OR UNTIL OUT
OF TICKETS STARTING WITH TICKET #1;
DO WHILE (TKTNUM(TKTEVT) NE . AND MAIL_MO(RPTMO)=0);
  IF BEGDATE(RPTMO)<=MAILDT(TKTEVT)<=ENDDATE(RPTMO)
    AND (TERMDT(TKTEVT)=. OR TERMDT(TKTEVT)>MAILDT(TKTEVT))
  THEN MAIL_MO(RPTMO)=1;
  TKTEVT=TKTEVT+1;
END;

*** FLAG TICKET ELIGIBLE FOR REPORTING MONTH STARTING WITH TICKET
#1;
*** BENE IS "TICKET ELIGIBLE" IN REPORTING MONTH IF ALL OF THE
FOLLOWING CONDITIONS ARE MEET:
1. TICKET MAILED ON OR BEFORE LAST DAY OF REPORTING MONTH
2. TICKET HAD EITHER NOT BEEN TERMINATED OR, IF TERMINATED,
TERMINATION DATE WAS LATER THAN THE MAIL DATE & ON OR
AFTER THE FIRST DAY OF THE REPORTING MONTH
3. BENE WAS OLDER THAN EXACTLY 18 & YOUNGER THAN FRA ON THE
LAST DAY OF REPORTING MONTH (SSA CONSIDERS A BENE TO BE AT
RETIREMENT AGE IN THE MONTH THEY REACH FRA)
4. BENES DATE OF DEATH WAS MISSING OR, IF NOT MISSING, IT WAS
AFTER THE LAST DAY OF THE REPORTING MONTH
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```

;
TKTEVT=1;
*** LOOP THRU TICKETS UNTIL ELIGIBLE IS FOUND OR UNTIL OUT OF
    TICKETS;
DO WHILE (TKTNUM(TKTEVT) NE . AND ELIGIBLE_MO(RPTMO)=0);
  IF MAILDT(TKTEVT) <= ENDDATE(RPTMO)
    AND MAILDT(TKTEVT) NE .
    AND 18 < AGE < FRA
    AND (DODTKT EQ . OR DODTKT >= BEGDATE(RPTMO))
    AND ((TERMDT(TKTEVT) > MAILDT(TKTEVT)
        AND TERMDT(TKTEVT) >= BEGDATE(RPTMO))
        OR TERMDT(TKTEVT) = .)
  THEN DO;
    ELIGTKTEVT = TKTNUM(TKTEVT);
    ELIGIBLE_MO(RPTMO) = 1;
    EVER_ELIG = 1;
  END;
  TKTEVT = TKTEVT + 1;
END;
*** END ELIGIBLE LOOP;

*** FLAG ACTIVE TICKET IN REPORTING MONTH;
*** BENES HAVE AN "ACTIVE TICKET" IF THE BENE WAS "TICKET
ELIGIBLE"
    IN REPORTING MONTH & EITHER OF THE FOLLOWING CONDITIONS ARE
    MEET:
    1. TICKET WAS ASSIGNED BEFORE THE 1ST DAY OF REPORTING MONTH
        & IT EITHER HAD NOT BEEN UNASSIGNED OR TERMINATED, OR IF
        UNASSIGNED OR TERMINATED, IT WAS ON OR AFTER THE FIRST DAY
        OF THE MONTH
    2. TICKET WAS ASSIGNED ON ANY DAY DURING THE MONTH
    ;

*** IF ELIGIBLE TICKET THIS REPORTING MONTH START ACTIVE LOOP;
IF ELIGIBLE_MO(RPTMO)=1 THEN DO;
  *** LOOP THROUGH ELIGIBLE ASSIGNMENTS;
  DO ASGN=ELIGTKTEVT TO NOE UNTIL(ACTIVE_MO(RPTMO)=1);
    IF((ASGNDT(ASGN) < BEGDATE(RPTMO) AND ASGNDT(ASGN) NE .)
      AND (UNASSGDT(ASGN) >= BEGDATE(RPTMO) OR UNASSGDT(ASGN) = .)
      AND (TERMDT(ASGN) >= BEGDATE(RPTMO) OR TERMDT(ASGN) = .)) OR
      BEGDATE(RPTMO) <= ASGNDT(ASGN) <= ENDDATE(RPTMO) THEN DO;
      ACTIVE_MO(RPTMO) = 1;
      EVER_ACTIVE = 1;
    END;
  END;
  *** END ASSIGNMENT LOOP;
END;
*** END ACTIVE LOOP;

*** DETERMINE PROVIDER AND PAYMENT TYPES FOR ACTIVE TICKETS IN
REPORTING MONTH;
IF ACTIVE_MO(RPTMO)=1 THEN DO;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
*** FLAG PAYMENT TYPE FOR ACTIVE TICKET;
PMTTYPE_MO(RPTMO)=PMTTYP(ASGN);

*** FLAG PROVIDER TYPE;
PROVTYPE_MO(RPTMO)=PROVTYPE(ASGN);

*** SVRAS;
IF PROVTYPE_MO(RPTMO)='V' THEN DO;
  EVER_VR=1;

  *** FLAG PROVIDER/PAYMENT TYPE COMBOS;
  SELECT(PMTTYPE_MO(RPTMO));
    WHEN ('V') DO; VR_TR_MO(RPTMO)=1; EVER_VR_TR=1; END;
    WHEN ('M') DO; VR_MO_MO(RPTMO)=1; EVER_VR_MO=1; END;
    WHEN ('O') DO; VR_OO_MO(RPTMO)=1; EVER_VR_OO=1; END;
    OTHERWISE DO;
      VR_TR_MO(RPTMO)=9;
      VR_MO_MO(RPTMO)=9;
      VR_OO_MO(RPTMO)=9;
    END;
  END;

END;

*** ENS;
ELSE IF PROVTYPE_MO(RPTMO)='E' THEN DO;
  EVER_EN=1;

  *** FLAG PROVIDER/PAYMENT TYPE COMBOS;
  SELECT(PMTTYPE_MO(RPTMO));
    WHEN('M') DO; EN_MO_MO(RPTMO)=1; EVER_EN_MO=1; END;
    WHEN('O') DO; EN_OO_MO(RPTMO)=1; EVER_EN_OO=1; END;
    OTHERWISE DO;
      EN_MO_MO(RPTMO)=9;
      EN_OO_MO(RPTMO)=9;
    END;
  END;

END;

ELSE PROVTYPE_MO(RPTMO)='X';
END;
*** END PROVIDER & PAYMENT TYPE LOOP;

*** SET MONTHLY EVER FLAGS;
IF EVER_ELIG=1 THEN ELIGIBLE_EV(RPTMO)=1;
ELSE ELIGIBLE_EV(RPTMO)=0;

IF EVER_ACTIVE=1 THEN ACTIVE_EV(RPTMO)=1;
ELSE ACTIVE_EV(RPTMO)=0;

IF EVER_VR=1 THEN VR_EV(RPTMO)=1;
ELSE VR_EV(RPTMO)=0;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
IF EVER_VR_TR=1 THEN VR_TR_EV(RPTMO)=1;
ELSE VR_TR_EV(RPTMO)=0;

IF EVER_VR_MO=1 THEN VR_MO_EV(RPTMO)=1;
ELSE VR_MO_EV(RPTMO)=0;

IF EVER_VR_OO=1 THEN VR_OO_EV(RPTMO)=1;
ELSE VR_OO_EV(RPTMO)=0;

IF EVER_EN=1 THEN EN_EV(RPTMO)=1;
ELSE EN_EV(RPTMO)=0;

IF EVER_EN_MO=1 THEN EN_MO_EV(RPTMO)=1;
ELSE EN_MO_EV(RPTMO)=0;

IF EVER_EN_OO=1 THEN EN_OO_EV(RPTMO)=1;
ELSE EN_OO_EV(RPTMO)=0;

*** FLAG BENES FIRST ASSIGNMENTS;
* FOR 1ST REPORTING MONTH -- JAN 02;
IF RPTMO=1 AND ACTIVE_EV(RPTMO)=1 THEN DO;
  FIRST_ASGN_MO(RPTMO)=1;
  IF VR_TR_MO(RPTMO)=1 OR VR_MO_MO(RPTMO)=1 OR VR_OO_MO(RPTMO)=1
    THEN FIRST_ASGN_VR_MO(RPTMO)=1;
  ELSE IF EN_MO_MO(RPTMO)=1 OR EN_OO_MO(RPTMO)=1
    THEN FIRST_ASGN_EN_MO(RPTMO)=1;

  *** FLAGS FOR REASSIGNMENT PROCESSING;
  FIRSTASGNDT=ASGNDT(ASGN);
  FIRSTASGNMO=RPTMO;
END;
* FOR SUBSEQUENT REPORTING MONTHS -- AFTER JAN 02;
ELSE IF ACTIVE_EV(RPTMO)=1 AND ACTIVE_EV(RPTMO-1)=0 THEN DO;
  FIRST_ASGN_MO(RPTMO)=1;
  IF VR_TR_MO(RPTMO)=1 OR VR_MO_MO(RPTMO)=1 OR VR_OO_MO(RPTMO)=1
    THEN FIRST_ASGN_VR_MO(RPTMO)=1;
  ELSE IF EN_MO_MO(RPTMO)=1 OR EN_OO_MO(RPTMO)=1
    THEN FIRST_ASGN_EN_MO(RPTMO)=1;

  *** FLAGS FOR REASSIGNMENT PROCESSING;
  FIRSTASGNDT=ASGNDT(ASGN);
  FIRSTASGNMO=RPTMO;
END;
*** END FIRST ASSIGNMENT LOOP;

*** COUNT DEACTIVATIONS;
*** A TICKET HAS BEEN DEACTIVATED WHEN BENE HAS HAD AN ASSIGNMENT
    IN OR BEFORE REPROTING MONTH & AN UNASSIGNED DATE IN THE
    REPORTING MONTH OR IF UNASSIGNED DATE IS MISSING HAS A
    TERMINATION DATE IN THE REPORTING MONTH;
IF ACTIVE_EV(RPTMO)=1 THEN DO UNASGN=1 TO NOE;
  IF (UNASSGDT(UNASGN) NE .
    AND BEGDATE(RPTMO)<=UNASSGDT(UNASGN)<=ENDDATE(RPTMO))
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
OR (UNASSGDT(UNASGN)=. AND TERMDT(UNASGN) NE .
    AND BEGDATE(RPTMO)<=TERMDT(UNASGN)<=ENDDATE(RPTMO))
THEN DO;
    DEACTIVE_MO(RPTMO)=DEACTIVE_MO(RPTMO)+1;
    IF VR_TR_MO(RPTMO)=1 OR VR_MO_MO(RPTMO)=1
    OR VR_OO_MO(RPTMO)=1 THEN
        DEACTIVE_VR_MO(RPTMO)=DEACTIVE_VR_MO(RPTMO)+1;
    ELSE IF EN_MO_MO(RPTMO)=1 OR EN_OO_MO(RPTMO)=1 THEN
        DEACTIVE_EN_MO(RPTMO)=DEACTIVE_EN_MO(RPTMO)+1;
END;
END;
*** END DEACTIVATIONS LOOP;

*** COUNT REASSIGNMENTS;
*** A TICKET HAS BEEN REASSIGNED WHEN BENE HAS HAD AN ASSIGNMENT
IN
    OR BEFORE REPORTING MONTH & A DEACTIVATION IN OR BEFORE
    REPORTING MONTH & HAS A SUBSEQUENT VALID ASSIGNMENT DATE
    WITHIN REPROTING MONTH THAT IS NOT EQUAL TO BENES FIRST
    ASSIGNMENT DATE;
IF ACTIVE_EV(RPTMO)=1 THEN DO I=FIRSTASGNMO TO RPTMO;
    IF DEACTIVE_MO(I)=1 THEN DO J=1 TO NOE;
        IF ASGNDT(J)>FIRSTASGNDT AND ASGNDT(J) NE .
            AND BEGDATE(RPTMO)<=ASGNDT(J)<=ENDDATE(RPTMO)
            AND (UNASSGDT(J)>BEGDATE(RPTMO) OR UNASSGDT(J)=.)
            AND (TERMDT(J)>BEGDATE(RPTMO) OR TERMDT(J)=.)
        THEN DO;
            REASSIGN_MO(RPTMO)=REASSIGN_MO(RPTMO)+1;
            IF VR_TR_MO(RPTMO)=1 OR VR_MO_MO(RPTMO)=1
            OR VR_OO_MO(RPTMO)=1 THEN
                REASSIGN_VR_MO(RPTMO)=REASSIGN_VR_MO(RPTMO)+1;
            ELSE IF EN_MO_MO(RPTMO)=1 OR EN_OO_MO(RPTMO)=1 THEN
                REASSIGN_EN_MO(RPTMO)=REASSIGN_EN_MO(RPTMO)+1;
        END;
    END;
END;
*** END REASSIGNMENTS LOOP;

END;
* END REPOTING MONTH LOOP;

* SET THE MINIMUM (FIRST) MAIL DATE;
MINMAIL=MIN(OF TKTMAILDDT1-TKTMAILDDT30);

DROP I J RPTMO TKTEVT ASGN UNASGN;

RUN;

PROC CONTENTS DATA=OUT.TKTTMF POSITION;
RUN;

*** ADDED IN TRF10 - THE PRINT CODE IS REMOVED FROM THIS ***
*** PROGRAM IN ORDER TO AVOID INADVERTANT LOSS OF PII. ***;
```

Appendix A.61
JCL/SAS Code: CRTKTFLG

```
*** THE CODE BELOW HAS BEEN MADE INTO A SEPARATE PROGRAM ***;  
*PROC PRINT DATA=OUT.TKTTMF(OBS=25);  
*RUN;  
  
%mend start;  
  
%start;  
  
ENDSAS;
```

Appendix A.62
JCL/SAS Code: TKTFINAL

```
//$2358FIN JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 06979
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.TKT.PRDLIB(TKTFINAL)
//*
//* DESCRIPTION: CREATE FINAL TICKET FILE
//*
//* DATE:       08/02/10 DAWN PHELPS
//* UPDATED:    02/06/12 NATALIE HAZELWOOD
//*****
//SAS        EXEC SAS9,
//          WORK='120000,60000'
//
//IN         DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.CRTKTFLG.SA.V1,DISP=SHR
//TKTBSD    DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKETBS.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL02   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET02.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL03   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET03.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL04   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET04.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL05   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET05.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL06   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET06.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL07   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET07.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL08   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET08.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL09   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET09.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TKTFL10   DD  DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.TICKET10.SA.V1,
//          DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//TEMP1     DD  DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN     DD  *
```

Appendix A.62

JCL/SAS Code: TKTFINAL

```

OPTIONS NOCENTER COMPRESS=BINARY MPRINT MACROGEN OBS=MAX
        DKROCOND=WARN;

TITLE1 "Ticket to Work";
TITLE2 "TKT TMF PHASE CREATION TRF10";
TITLE3;

        /* change endyr and endmmn for each run */
%let begyr=2002;
%let endyr=2010; /* change as needed */
%let endmn=12;   /* change as needed */

        /* step to assign macro variables to handle time series data */
%macro start;
%let k=1;
%do i=&begyr %to &endyr;
    %if &i<2000 %then %let yr=%eval(&i-1900);
    %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
    %else %let yr=%eval(&i-2000);
    %if &i=&endyr %then %let emn=%eval(&endmn);
    %else %let emn=12;
    %do j=1 %to &emn;
        %if &j<10 %then %let mn=0%eval(&j);
        %else %let mn=%eval(&j);
        %let x&k=%eval(&yr)%eval(&mn);
        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

DATA TKTBS.TICKETBS(DROP=ACTIVE_EV: ACTIVE_MO: DEACTIVE_EN_MO:
                    DEACTIVE_MO: DEACTIVE_VR_MO: ELIGIBLE_EV:
                    ELIGIBLE_MO: EN_EV: EN_MO_EV: EN_MO_MO:
                    EN_OO_EV: EN_OO_MO: FIRST_ASGN_EN_MO:
                    FIRST_ASGN_MO: FIRST_ASGN_VR_MO: MAIL_MO:
                    PMTTYE_MO: PROVTYPE_MO: REASSIGN_EN_MO:
                    REASSIGN_MO: REASSIGN_VR_MO: TITLE_MO:
                    VR_EV: VR_MO_EV: VR_MO_MO: VR_OO_EV: VR_OO_MO:
                    VR_TR_EV: VR_TR_MO: TKTEIN: CINUSE_MO:
                    INUSE_MO:)
%DO I=&BEGYR. %TO &ENDYR.;
    %LET J=%SUBSTR(&I,3,2);
    TKTFL&J..TICKET&J.(KEEP=COSSN
                        ACTIVE_EV&J: ACTIVE_MO&J: DEACTIVE_EN_MO&J:
                        DEACTIVE_MO&J:                                DEACTIVE_VR_MO&J:
ELIGIBLE_EV&J:
                        ELIGIBLE_MO&J:                                EN_EV&J:                                EN_MO_EV&J:
EN_MO_MO&J:
                        EN_OO_EV&J: EN_OO_MO&J: FIRST_ASGN_EN_MO&J:
                        FIRST_ASGN_MO&J:                                FIRST_ASGN_VR_MO&J:
MAIL_MO&J:
                        PMTTYE_MO&J: PROVTYPE_MO&J: REASSIGN_EN_MO&J:

```

Appendix A.62
JCL/SAS Code: TKTFINAL

```
REASSIGN_MO&J: REASSIGN_VR_MO&J: TITLE_MO&J:
VR_EV&J: VR_MO_EV&J: VR_MO_MO&J: VR_OO_EV&J:
VR_OO_MO&J: VR_TR_EV&J: VR_TR_MO&J:

INUSE_MO&J:)
  %END;
;
SET IN.TKTTMF(DROP=_LABEL_
              _NAME_
              AGE
              BEGDATE:
              DOBRETIRE
              ELIGTKTEVT
              ENDDATE:
              FIRSTASGNDT
              FIRSTASGNMO
              INCLNT
              MO
              NUMBER:
              RD1:
              RD2:
              TRF06_:
              TRF09_:
              TITLE_MO00:
              TITLE_MO01:
              TITLE_MO11:
              TITLE_MO12:
              TITLE_MO8:
              TITLE_MO9:
              TKTDOB
              YR)
;

LABEL
%DO I=1 %TO &TOT;
  ACTIVE_EV&&X&I="Ever assigned ticket as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  ACTIVE_MO&&X&I="Ticket active (assigned) in &&X&I"
%END;
COSSN="Beneficiary's Own SSN"
%DO I=1 %TO &TOT;
  DEACTIVE_EN_MO&&X&I="Ticket unassigned from EN in &&X&I"
%END;
%DO I=1 %TO &TOT; DEACTIVE_MO&&X&I="Ticket unassigned in &&X&I" %END;
%DO I=1 %TO &TOT;
  DEACTIVE_VR_MO&&X&I="Ticket unassigned from SVRA in &&X&I" %END;
DOBTKT="DATE OF BIRTH USED TO DETERMINE TICKET ELIGIBILITY"
DODTKT="DATE OF DEATH USED TO DETERMINE TICKET ELIGIBILITY"
%DO I=1 %TO 30; DUNS&I.="DUNS OCCURENCE &I." %END;
%DO I=1 %TO &TOT;
  ELIGIBLE_EV&&X&I="Ever eligible for a ticket as of &&X&I" %END;
%DO I=1 %TO &TOT;
  ELIGIBLE_MO&&X&I="Eligible for a ticket in &&X&I" %END;
```

Appendix A.62
JCL/SAS Code: TKTFINAL

```
%DO I=1 %TO &TOT; EN_EV&&X&I="Ever assigned ticket to EN as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  EN_MO_EV&&X&I="Ever assigned Ticket to EN-MO as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  EN_MO_MO&&X&I="Ticket assigned to EN-MO in &&X&I"
%END;
%DO I=1 %TO &TOT;
  EN_OO_EV&&X&I="Ever assigned Ticket to EN-OO as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  EN_OO_MO&&X&I="Ticket assigned to EN-OO in &&X&I"
%END;
EVER_ACTIVE="Ticket Ever Assigned as of date of data extract"
EVER_ELIG="Ever eligible as of date of data extract"
EVER_EN="Ever assigned ticket to EN as of data extract"
EVER_EN_MO="Ever assigned ticket to EN:MO as of data extract"
EVER_EN_OO="Ever assigned ticket to EN:OO as of data extract"
EVER_VR="Ever assigned ticket to SVRA as of last month"
EVER_VR_MO="Ever assigned ticket to SVRA:MO as of data extract"
EVER_VR_OO="Ever assigned ticket to SVRA:OO as of data extract"
EVER_VR_TR="Ever assigned ticket to SVRA:TR as of data extract"
%DO I=1 %TO &TOT;
  FIRST_ASGN_EN_MO&&X&I="First assigned Ticket to EN in &&X&I"
%END;
%DO I=1 %TO &TOT;
  FIRST_ASGN_MO&&X&I="First assigned Ticket in &&X&I"
%END;
%DO I=1 %TO &TOT;
  FIRST_ASGN_VR_MO&&X&I="First assigned Ticket to VR in &&X&I"
%END;
FRA="BENEFICIARY'S FULL RETIREMENT AGE"
%DO I=1 %TO &TOT;
  MAIL_MO&&X&I="TICKET MAILED FLAG FOR &&X&I."
%END;
MINMAIL='FIRST TICKET MAIL DATE'
NOE='NUMBER OF ENTRIES'
%DO I=1 %TO 30; PMTTYPER&I.="PAYMENT TYPE NUMBER &I." %END;
%DO I=1 %TO &TOT;
  PMTTYPER_MO&&X&I="TICKET PAYMENT TYPE in &&X&I"
%END;
%DO I=1 %TO 30; PROVTYPE&I.="PROVIDER TYPE NUMBER &I." %END;
%DO I=1 %TO &TOT;
  PROVTYPE_MO&&X&I="TICKET PROVIDER TYPE in &&X&I"
%END;
%DO I=1 %TO &TOT;
  REASSIGN_EN_MO&&X&I="REASSIGNED TICKET TO EN in &&X&I"
%END;
%DO I=1 %TO &TOT;
  REASSIGN_MO&&X&I="REASSIGNED TICKET in &&X&I"
%END;
%DO I=1 %TO &TOT;
```

Appendix A.62
JCL/SAS Code: TKTFINAL

```
REASSIGN_VR_MO&&X&I="REASSIGNED TICKET TO VR in &&X&I"
%END;
%DO I=1 %TO &TOT;
  TITLE_MO&&X&I="TITLE in &&X&I"
%END;
%DO I=1 %TO 30; TKTASGNDDT&I.="TICKET ASSIGNMENT DATE NUMBER &I."
%END;
%DO I=1 %TO 5; TKTEIN&I.="TICKET EIN NUMBER &I." %END;
%DO I=1 %TO 30; TKTMAILDDT&I.="TICKET MAIL DATE NUMBER &I." %END;
%DO I=1 %TO 30; TKTTERMDDT&I.="TICKET TERMINATION DATE NUMBER &I."
%END;
%DO I=1 %TO 30; TKTUNASGDT&I.="TICKET UNASSIGNED DATE NUMBER &I."
%END;
TRF1='BENEFICIARY WAS IN TRF 1'
TRF2='BENEFICIARY WAS IN TRF 2'
TRF05='BENEFICIARY WAS IN TRF05'
TRF06='BENEFICIARY WAS IN TRF06'
TRF07='BENEFICIARY WAS IN TRF07'
TRF08='BENEFICIARY WAS IN TRF08'
TRF09='BENEFICIARY WAS IN TRF09'
TRF10='BENEFICIARY WAS IN TRF10'
%DO I=1 %TO &TOT;
  VR_EV&&X&I="Ever assigned ticket to SVRA as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_MO_EV&&X&I="Ever assigned ticket to EN:MO as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_MO_MO&&X&I="TICKET ASSIGNED TO VR:MO IN &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_OO_EV&&X&I="Ever assigned ticket to SVRA:OO as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_OO_MO&&X&I="TICKET ASSIGNED TO VR:OO IN &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_TR_EV&&X&I="Ever assigned ticket to SVRA:TR as of &&X&I"
%END;
%DO I=1 %TO &TOT;
  VR_TR_MO&&X&I="TICKET ASSIGNED TO SVRA:TR IN &&X&I"
%END;
%DO I=1 %TO &TOT;
  INUSE_MO&&X&I="Ticket In-Use in &&X&I"
%END;
;
RUN;

PROC CONTENTS DATA=TKTB.S.TICKETBS;
RUN;

%DO I=2 %TO 9;
  PROC CONTENTS DATA=TKTFLO&I..TICKET0&I.;
```

Appendix A.62
JCL/SAS Code: TKTFINAL

```
RUN;  
%END;  
  
*** ADD NEW LOOP FOR TRF11 ***;  
PROC CONTENTS DATA=TKTFL10.TICKET10;  
RUN;  
%mend start;  
  
%start;  
  
ENDSAS;
```

Appendix A.63
JCL/SAS Code: DCFERNEX

```
// $2358DCF JOB (12510000,T715,,SAS,,ITC9TH),2358HAZE,CLASS=T,
// MSGCLASS=1,MSGLEVEL=(1,1),REGION=4096K,NOTIFY=$2358
/*SSAMAIN SASBASIC
//*****
/* PROJECT: BUILD TRF10 (06979)
/* PROJECT
/* DIRECTOR: LAURA KOSAR
/*
/* PROGRAM: OPDR.TG.PRD.ETTW.$2358.TRF10.DCFANN.PRDLIB(DCFERNEX)
/*
/* DESCRIPTION: CREATES EXTRACT OF DCF EARNINGS TABLES
/* TO OBTAIN THE EARNINGS DATA FOR THE ANNUAL FILES
/*
/* DATE: 06/24/2010 DAWN PHELPS
/* UPDATED: 12/1/2011 NATALIE HAZELWOOD
//*****
//S1 EXEC SAS9,LOAD='DBP8.DB2.SDSNLOAD',SORT='15000',
// WORK='15000,15000',COND=EVEN
//ALE DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFALERN.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=TSILO,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HALE DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHALER.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//T16E DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT16ER.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HT16 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHT16E.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//T2ER DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT2ER.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//HT2E DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHT2E.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
//T2WK DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT2WKD.SA.V1,
// DISP=(,CATLG,DELETE),
// UNIT=AFF=ALE,
// VOL=(,,255),
// DCB=(BLKSIZE=6144,DSORG=PS,LRECL=6144,RECFM=FS)
```

Appendix A.63
JCL/SAS Code: DCFERNEX

```
//HT2W DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFHT2WK.SA.V1 ,  
// DISP=( ,CATLG,DELETE) ,  
// UNIT=AFF=ALE ,  
// VOL=( , , ,255) ,  
// DCB=(BLKSIZE=6144 ,DSORG=PS ,LRECL=6144 ,RECFM=FS )  
//SYSIN DD *
```

```
options nocenter;
```

```
LIBNAME ALE TAPE;  
LIBNAME HALE TAPE;  
LIBNAME T16E TAPE;  
LIBNAME HT16 TAPE;  
LIBNAME T2ER TAPE;  
LIBNAME HT2E TAPE;  
LIBNAME T2WK TAPE;  
LIBNAME HT2W TAPE;  
LIBNAME MDCF DB2 SSID=DBP8 SCHEMA=MDCF ;
```

```
* CREATE SAS EXTRACT OF ALLEGED EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE ALE.ALLGERNG AS  
SELECT * FROM MDCF.ALLGERNG;  
QUIT;  
RUN;
```

```
* CREATE SAS EXTRACT OF HISTORY ALLEGED EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE HALE.HALLGERN AS  
SELECT * FROM MDCF.HALLGERN;  
QUIT;  
RUN;
```

```
* CREATE SAS EXTRACT OF T16 EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE T16E.T16ERNGS AS  
SELECT * FROM MDCF.T16ERNGS;  
QUIT;  
RUN;
```

```
* CREATE SAS EXTRACT OF HISTORY T16 EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE HT16.HT16ERNG AS  
SELECT * FROM MDCF.HT16ERNG;  
QUIT;
```

```
* CREATE SAS EXTRACT OF T2 EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE T2ER.T2ERNGS AS  
SELECT * FROM MDCF.T2ERNGS;  
QUIT;  
RUN;
```

Appendix A.63
JCL/SAS Code: DCFERNEX

```
* CREATE SAS EXTRACT OF HISTORY T2 EARNINGS GROUP DATA;  
PROC SQL;  
CREATE TABLE HT2E.HT2ERNGS AS  
SELECT * FROM MDCF.HT2ERNGS;  
QUIT;  
RUN;
```

```
* CREATE SAS EXTRACT OF T2 WORK GROUP DATA;  
PROC SQL;  
CREATE TABLE T2WK.T2WKDETN AS  
SELECT * FROM MDCF.T2WKDETN  
QUIT;  
RUN;
```

```
* CREATE SAS EXTRACT OF HISTORY T2 WORK GROUP DATA;  
PROC SQL;  
CREATE TABLE HT2W.HT2WKDET AS  
SELECT * FROM MDCF.HT2WKDET;  
QUIT;  
RUN;
```

```
PROC CONTENTS DATA=ALE.ALLGERNG;  
RUN;  
PROC CONTENTS DATA=HALE.HALLGERN;  
RUN;  
PROC CONTENTS DATA=T16E.T16ERNGS;  
RUN;  
PROC CONTENTS DATA=HT16.HT16ERNG;  
RUN;  
PROC CONTENTS DATA=T2ER.T2ERNGS;  
RUN;  
PROC CONTENTS DATA=HT2E.HT2ERNGS;  
RUN;  
PROC CONTENTS DATA=T2WK.T2WKDETN;  
RUN;  
PROC CONTENTS DATA=HT2E.HT2WKDET;  
RUN;
```

Appendix A.64
JCL/SAS Code: ALGEARN

```

//$2358ALG JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 06979
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.DCFANN.PRDLIB(ALGEARN)
//*
//* DESCRIPTION: BUILDS THE YMM EARNINGS VARIABLES FROM THE DCF
//*              ALLEGED EARNINGS TABLE
//*
//* DATE:       08/03/10 DAWN PHELPS
//* UPDATED:    12/08/11 NATALIE HAZELWOOD
//*****
//SAS          EXEC SAS9,
//              WORK='120000,60000'
//
//
//IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFALERN.SA.V1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,
//      DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//      SPACE=(CYL,(3000,3000),RLSE)
//TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP3 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS LS=132 COMPRESS=YES MPRINT MACROGEN OBS=MAX;
  /* change endyr and endmmn for each run */
%let begyr=2000;
%let endyr=2010; /* change as needed */
%let endmn=12;   /* change as needed */

  /* step to assign macro variables to handle time series data */
%LET NUMYRS=%EVAL(&ENDYR.-&BEGYR.+1);

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
  %end;
%end;

```

Appendix A.64
JCL/SAS Code: ALGEARN

```
        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

*****                ALLEGED                EARNINGS                PROCESSING
*****;

DATA ALLERN;
    SET IN1.ALLGERNG;

    * IF THE EARNINGS DATE IS IN A REPORTING YEAR THEN KEEP;
    WHERE ERNGS_DT<='31DEC2010'D AND CID='00';

    * ATTACH FORMAT TO THE ERNGS_DT FIELD SO IT WILL APPEAR AS Yymm;
    ATTRIB ERNGS_DT FORMAT=YymmN4.;

RUN;

PROC SORT DATA=ALLERN;
    BY COSSN;
RUN;

* CREATE ONE OBSERVATION PER SSN WITH Yymm VARIABLES CONTAINING THE
  ALLEGED EARNINGS AMOUNT;
PROC TRANSPOSE DATA=ALLERN OUT=TEMP1.ALLERN PREFIX=ALLGAMT;
    BY COSSN;
    ID ERNGS_DT;
    VAR ALLGD_AMT;
RUN;

DATA OUT.ALGEARN;
    SET TEMP1.ALLERN;
    ARRAY OVARS {1,&NUMYRS.,12}
        %DO I=1 %TO &TOT; ALLGAMT&&X&I %END;;
    LABEL %DO I=1 %TO &TOT;
        ALLGAMT&&X&I="Earnings Alleged Amt &&X&I" %END;
RUN;

PROC CONTENTS DATA=OUT.ALGEARN;
RUN;

PROC PRINT DATA=OUT.ALGEARN(OBS=25);
RUN;

%MEND START;
%START;
ENDSAS;
```

Appendix A.65
JCL/SAS Code: T16EARN

```

//$2358T16 JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:    LAURA KOSAR
//*
//* PROGRAM:     OPDR.TG.PRD.ETTW.$2358.TRF10.ANN.PRDLIB(T16EARN)
//*
//* DESCRIPTION: BUILDS THE YMM EARNINGS VARIABLES FROM THE DCF
//          T16EARN FILE
//*
//* DATE:        08/03/10 DAWN PHELPS
//* UPDATED:     12/1/11 NATALIE HAZELWOOD
//*****
//SAS          EXEC SAS9,
//          WORK='120000,60000'
//
//
//IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT16ER.SA.V1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP3 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS LS=132 COMPRESS=YES MPRINT MACROGEN OBS=MAX;
/* change endyr and endmmn for each run */
%let begyr=2000;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */
%LET NUMYRS=%EVAL(&ENDYR.-&BEGYR.+1);

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %if %eval(&i-2000)<10 %then %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
  %end;
%end;

```

Appendix A.65
JCL/SAS Code: T16EARN

```

        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

*****                                T16                                EARNINGS                                PROCESSING
*****;

DATA TEMP3.T16ERN;
    SET IN1.T16ERNGS;

    * IF THE EARNINGS DATE IS IN A REPORTING YEAR THEN KEEP;
    WHERE ERNGS_DT<='31DEC2010'D AND CID='00';

    * ATTACH FORMAT TO THE ERNGS_DT FIELD SO IT WILL APPEAR AS YYMM;
    ATTRIB ERNGS_DT FORMAT=YYMMN4.;

RUN;

PROC SORT DATA=TEMP3.T16ERN;
    BY COSSN;

RUN;

* CREATE ONE OBSERVATION PER SSN WITH YYMM VARIABLES CONTAINING EACH
  OR THE EARNINGS AMOUNTS;
%MACRO T16EARN(ERNVAR,ERNPREF);
    PROC          TRANSPOSE          DATA=TEMP3.T16ERN          OUT=TEMP3.&ERNPREF.
PREFIX=&ERNPREF.;
        BY COSSN;
        ID ERNGS_DT;
        VAR &ERNVAR.;
    RUN;
%MEND T16EARN;

%T16EARN(STUD_EINCMXCL_AMT,T16EXLAMT);
%T16EARN(WRK_EXP_AMT,T16EXPAMT);
%T16EARN(GRS_AMT,T16GRSAMT);
%T16EARN(SE_NET_AMT,T16NETAMT);
%T16EARN(PASS_AMT,T16PASAMT);
%T16EARN(VRFD_IND,T16VERIND);
%T16EARN(SE_VRFD_IND,T16SEVERIND);

*****                                COMINE                                ALL                                EARNINGS
*****;

DATA OUT.T16EARN;
    MERGE TEMP3.T16EXLAMT
          TEMP3.T16EXPAMT
          TEMP3.T16GRSAMT
          TEMP3.T16NETAMT
          TEMP3.T16PASAMT
          TEMP3.T16VERIND
          TEMP3.T16SEVERIND
    ;

```

Appendix A.65
JCL/SAS Code: T16EARN

BY COSSN;

* THIS ARRAY WILL ENSURE THAT THERE IS A VARIABLE FOR EVERY EARNINGS

TYPE FOR EVERY YYMM COMBINATION. ALTHOUGH UNLIKELY IT IS POSSIBLE THAT THERE WILL BE NO BENES WHO HAD A CERTAIN EARNINGS TYPE IN A GIVEN MONTH -- IF THIS WERE TO HAPPEN THEN THERE WOULD

BE NO YYMM VARIABLE FOR THAT EARNING TYPE SINCE THE PROC TRANSPOSE IN THE PREVIOUS STEP ONLY CREATES VARIABLES FOR DATA IT ENCOUNTERS

THIS IS A MULTIDIMENSIONAL ARRAY. THE FIRST DIMENSION REPRESENTS

THE NUMBER OF VARIABLE GROUPS, THE SECOND DIMENSION REPRESENTS THE NUMBER OF YEARS BEING REPORTED, THE THIRD DIMENSION REPRESENTS THE NUMBER OF MONTHS WITHIN EACH YEAR. THERE ARE 7 VARIABLE GROUPS WITH 12 MONTHLY OCCURENCES FOR &NUMYRS. - SEE BEGIN OF PROGRAM TO SEE HOW MANY YEARS ARE BEGIN REPORTED;

ARRAY OVARS {5,&NUMYRS.,12}

%DO I=1 %TO &TOT; T16EXLAMT&&X&I %END;

%DO I=1 %TO &TOT; T16EXPAMT&&X&I %END;

%DO I=1 %TO &TOT; T16GRSAMT&&X&I %END;

%DO I=1 %TO &TOT; T16NETAMT&&X&I %END;

%DO I=1 %TO &TOT; T16PASAMT&&X&I %END;

;

ARRAY OCVAR {2,&NUMYRS.,12}

%DO I=1 %TO &TOT; T16VERIND&&X&I %END;

%DO I=1 %TO &TOT; T16SEVERIND&&X&I %END;

;

LABEL %DO I=1 %TO &TOT;

T16EXLAMT&&X&I="T16 Student Exclusion Amt &&X&I" %END;

%DO I=1 %TO &TOT;

T16EXPAMT&&X&I="T16 Work Expense Amt &&X&I" %END;

%DO I=1 %TO &TOT;

T16GRSAMT&&X&I="T16 Earnings Gross Amt &&X&I" %END;

%DO I=1 %TO &TOT;

T16NETAMT&&X&I="T16 Self-Employment Net Income Amt &&X&I" %END;

%DO I=1 %TO &TOT;

T16PASAMT&&X&I="T16 PASS Amt &&X&I" %END;

%DO I=1 %TO &TOT;

T16VERIND&&X&I="T2 Gross Earnings Amount Verified Switch &&X&I" %END;

%DO I=1 %TO &TOT;

T16SEVERIND&&X&I=

"T16 Self Reported Earnings Verified Switch &&X&I" %END;

;

RUN;

PROC CONTENTS DATA=OUT.T16EARN;

RUN;

Appendix A.65
JCL/SAS Code: T16EARN

```
PROC PRINT DATA=OUT.T16EARN(OBS=25);  
RUN;
```

```
%MEND START;  
%START;  
ENDSAS;
```

Appendix A.66
JCL/SAS Code: T2WKDETN

```

//$2358T2W JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 06979
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.DCFANN.PRDLIB(T2WKDETN)
//*
//* DESCRIPTION: BUILDS THE YMM EARNINGS VARIABLES FROM THE DCF
//          T2WKDETN TABLE
//*
//* DATE:       08/03/10 DAWN PHELPS
//* UPDATED:    12/08/11 NATALIE HAZELWOOD
//*****
//*
//SAS          EXEC SAS9,
//          WORK='120000,60000'
//*
//*
//IN1 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT2WKD.SA.V1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//TEMP1 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP3 DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS LS=132 COMPRESS=YES MPRINT MACROGEN OBS=MAX;
/* change endyr and endmmn for each run */
%let begyr=1994;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */
%LET NUMYRS=%EVAL(&ENDYR.-&BEGYR.+1);

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %if %eval(&i-2000)<10 %then %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
  %end;
%end;

```

Appendix A.66
JCL/SAS Code: T2WKDET

```
        %let k=%eval(&k+1);
    %end;
%end;
%let tot=%eval(&k-1);

*****                                T2                                WORK                                PROCESSING
*****;

DATA TEMP2.T2WKDET;
    SET IN1.T2WKDET;

    * IF THE EARNINGS DATE IS IN A REPORTING YEAR THEN KEEP;
    WHERE '01JAN1994'D<=ERNGS_DT<='31DEC2010'D AND CID='00';

    * ATTACH FORMAT TO THE ERNGS_DT FIELD SO IT WILL APPEAR AS Y4MM;
    ATTRIB ERNGS_DT FORMAT=Y4MMN4.;

RUN;

PROC SORT DATA=TEMP2.T2WKDET;
    BY COSSN ERNGS_DT DESCENDING LU_TS;
RUN;

PROC SORT DATA=TEMP2.T2WKDET NODUPKEYS;
    BY COSSN ERNGS_DT;
RUN;

* CREATE ONE OBSERVATION PER SSN WITH Y4MM VARIABLES CONTAINING EACH
  OR THE EARNINGS AMOUNTS;
%MACRO T2WKD(ERNVAR,ERNPREF);
    PROC TRANSPOSE DATA=TEMP2.T2WKDET OUT=TEMP2.&ERNPREF.
    PREFIX=&ERNPREF.;
        BY COSSN;
        ID ERNGS_DT;
        VAR &ERNVAR.;
    RUN;

%MEND T2WKD;

%T2WKD(SPCL_CDNS_AMT,T2CDNAMT);
%T2WKD(WRK_EXP_AMT,T2EXPAMT);
%T2WKD(SBDY_AMT,T2SBDYAMT);
%T2WKD(SE_UNBIZEXP_AMT,T2UBEAMT);
%T2WKD(WRK_DET_CD,TWPDATA);
%T2WKD(FRAUD_VRFD_SW,T2FRAUDVER);

DATA OUT.T2WKDET;
    MERGE TEMP2.T2CDNAMT
          TEMP2.T2EXPAMT
          TEMP2.T2SBDYAMT
          TEMP2.T2UBEAMT
          TEMP2.TWPDATA
          TEMP2.T2FRAUDVER
```

Appendix A.66
JCL/SAS Code: T2WKDETN

```

;
BY COSSN;

DROP  _NAME_
      _LABEL_
;

ARRAY OVARS {4,&NUMYRS.,12}
      %DO I=1 %TO &TOT; T2CDNAMT&&X&I %END;
      %DO I=1 %TO &TOT; T2EXPAMT&&X&I %END;
      %DO I=1 %TO &TOT; T2SBDYAMT&&X&I %END;
      %DO I=1 %TO &TOT; T2UBEAMT&&X&I %END;
;

ARRAY OCVARs {2,&NUMYRS.,12}
      %DO I=1 %TO &TOT; TWPDATA&&X&I %END;
      %DO I=1 %TO &TOT; T2FRAUDVER&&X&I %END;
;

LABEL %DO I=1 %TO &TOT;
      T2CDNAMT&&X&I="T2 Special Condition Amt &&X&I" %END;
      %DO I=1 %TO &TOT;
      T2EXPAMT&&X&I="T2 Work Expense Amt &&X&I" %END;
      %DO I=1 %TO &TOT;
      T2SBDYAMT&&X&I="T2 Earnings Subsidy Amt &&X&I" %END;
      %DO I=1 %TO &TOT;
      T2UBEAMT&&X&I="T2 Self-Employment Unpaid Business Exp Amt
&&X&I"
      %END;
      %DO I=1 %TO &TOT;
      TWPDATA&&X&I="T2 Trial Work Period Data &&X&I" %END;
      %DO I=1 %TO &TOT;
      T2FRAUDVER&&X&I="T2 Fraud Verified Switch &&X&I" %END;
;

RUN;

PROC CONTENTS DATA=OUT.T2WKDETN;
RUN;

PROC PRINT DATA=OUT.T2WKDETN(OBS=25);
RUN;

%MEND START;
%START;
ENDSAS;
```

Appendix A.67
JCL/SAS Code: T2EARN

```

//$2358T2   JOB (12510000,T715,,SAS,,ITC9FL),2358HAZE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$2358
//*****
//* PROJECT: BUILD TRF10 (06979)
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.$2358.TRF10.DCFANN.PRDLIB(T2EARN)
//*
//* DESCRIPTION: BUILDS THE YMM EARNINGS VARIABLES FROM THE DCF
//          T2EARN TABLE
//*
//* DATE:       08/03/10 DAWN PHELPS
//* UPDATED:    12/2/11 NATALIE HAZELWOOD
//*
//*****
//*
//SAS       EXEC SAS9,
//          WORK='120000,60000'
//*
//*
//IN1      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.DCFT2ER.SA.V1,DISP=SHR
//OUT      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          SPACE=(CYL,(3000,3000),RLSE)
//TEMP1    DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP2    DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEMP3    DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN    DD *

OPTIONS LS=132 COMPRESS=YES MPRINT MACROGEN OBS=MAX;
  /* change endyr and endmmn for each run */
%let begyr=1994;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

  /* step to assign macro variables to handle time series data */
%LET NUMYRS=%EVAL(&ENDYR.-&BEGYR.+1);

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %if %eval(&i-2000)<10 %then %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
  %end
%end

```

Appendix A.67
JCL/SAS Code: T2EARN

```
%let x&k=%eval(&yr)%eval(&mn);
%let k=%eval(&k+1);
%end;
%end;
%let tot=%eval(&k-1);

***** T2 EARNINGS PROCESSING
*****;

DATA TEMP2.T2ERN;
  SET IN1.T2ERN;

  * IF THE EARNINGS DATE IS IN A REPORTING YEAR THEN KEEP;
  WHERE '01JAN1994'D<=ERNGS_DT<='31DEC2010'D AND CID='00';

  * ATTACH FORMAT TO THE ERNGS_DT FIELD SO IT WILL APPEAR AS YYMM;
  ATTRIB ERNGS_DT FORMAT=YYMMN4.;

RUN;

PROC SORT DATA=TEMP2.T2ERN;
  BY COSSN;
RUN;

* CREATE ONE OBSERVATION PER SSN WITH YYMM VARIABLES CONTAINING EACH
  OR THE EARNINGS AMOUNTS;
%MACRO T2EARN(ERNVAR,ERNPREF);
PROC TRANSPOSE DATA=TEMP2.T2ERN OUT=TEMP2.&ERNPREF. PREFIX=&ERNPREF.;
  BY COSSN;
  ID ERNGS_DT;
  VAR &ERNVAR.;
RUN;

%MEND T2EARN;

%T2EARN(GRS_AMT,T2GRSAMT);
%T2EARN(SE_NET_AMT,T2NETAMT);
%T2EARN(SE_HRS,T2SEHRS);
%T2EARN(VRFD_IND,T2VERIND);
%T2EARN(SE_VRFD_IND,T2SEVERIND);

DATA OUT.T2EARN;
  MERGE TEMP2.T2GRSAMT
        TEMP2.T2NETAMT
        TEMP2.T2SEHRS
        TEMP2.T2VERIND
        TEMP2.T2SEVERIND
        ;
  BY COSSN;

  DROP _NAME_
        _LABEL_
        ;
```

Appendix A.67
JCL/SAS Code: T2EARN

```
* THIS ARRAY WILL ENSURE THAT THERE IS A VARIABLE FOR EVERY
EARNINGS
TYPE FOR EVERY YYMM COMBINATION. ALTHOUGH UNLIKELY IT IS
POSSIBLE THAT THERE WILL BE NO BENES WHO HAD A CERTAIN EARNINGS
TYPE IN A GIVEN MONTH -- IF THIS WERE TO HAPPEN THEN THERE
WOULD
BE NO YYMM VARIABLE FOR THAT EARNING TYPE SINCE THE PROC
TRANSPPOSE
IN THE PREVIOUS STEP ONLY CREATES VARIABLES FOR DATA IT
ENCOUNTERS

THIS IS A MULTIDIMENSIONAL ARRAY. THE FIRST DIMENSION
REPRESENTS
THE NUMBER OF VARIABLE GROUPS, THE SECOND DIMENSION REPRESENTS
THE NUMBER OF YEARS BEING REPORTED, THE THIRD DIMENSION
REPRESENTS THE NUMBER OF MONTHS WITHIN EACH YEAR. THERE ARE 5
VARIABLE GROUPS WITH 12 MONTHLY OCCURENCES FOR &NUMYRS. - SEE
BEGIN OF PROGRAM TO SEE HOW MANY YEARS ARE BEGIN REPORTED;
ARRAY OVARS {3,&NUMYRS.,12}
  %DO I=1 %TO &TOT; T2GRSAMT&&X&I %END;
  %DO I=1 %TO &TOT; T2NETAMT&&X&I %END;
  %DO I=1 %TO &TOT; T2SEHRS&&X&I %END;
  ;
ARRAY OCVAR {2,&NUMYRS.,12}
  %DO I=1 %TO &TOT; T2VERIND&&X&I %END;
  %DO I=1 %TO &TOT; T2SEVERIND&&X&I %END;
  ;

LABEL %DO I=1 %TO &TOT;
  T2GRSAMT&&X&I="T2 Earnings Gross Amt &&X&I" %END;
  %DO I=1 %TO &TOT;
  T2NETAMT&&X&I="T2 Self-Employment Net Income Amt &&X&I"
%END;

  %DO I=1 %TO &TOT;
  T2SEHRS&&X&I="T2 Self-Employment Hours &&X&I" %END;
  %DO I=1 %TO &TOT;
  T2VERIND&&X&I="T2 Gross Earnings Amount Verified Switch &&X&I" %END;
  %DO I=1 %TO &TOT;
  T2SEVERIND&&X&I="T2 Self Reported Earnings Verified Switch &&X&I"
%END;

  ;

RUN;

PROC CONTENTS DATA=OUT.T2EARN;
RUN;

PROC PRINT DATA=OUT.T2EARN(OBS=25);
RUN;

%MEND START;
%START;
ENDSAS;
```

Appendix A.68
JCL/SAS Code: LDWSSI

```

//#2127T16 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*      *-----*
//*      | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWSSI) |
//*      *-----*
//*
//* CREATE REVISED SSI INDICATORS
//* REVISION FOR TRF10
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* CONSTRUCT SSI LEFT DUE TO WORK INDICATORS
//* MODIFIED      12/21/2011 JEREMY PAGE FOR TRF10
//*****
//*
//JS010    EXEC SAS9,
//          WORK='120000,60000'
//*
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//DEM      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//OUT      DD DSN=OPDR.TG.PRD.ETTW.#2127.LDWSSI.Y2010.SSD,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN    DD *
OPTIONS NOCENTER COMPRESS=BINARY OBS=MAX;
%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

```

Appendix A.68
JCL/SAS Code: LDWSSI

```
DATA OUT.LDWSSI (KEEP=SSN LDWSSI:);
MERGE DEM.DMLDW
      (IN=D KEEP=SSN DOBBEST DODBEST)
IN1.LONG (IN=L KEEP=SSN PST: EIN: UIN: MFT)
IN2.T16EARN (RENAME=(COSSN=SSN));
BY SSN;
IF L ;
/* use dobbest from demo to compute retiredate */

/* compute retirement date for comparison */
/* if b'day is january 1st - refer to previous year */
/* sas9 offers "sameday" which simplifies the code below */
/* basis for code is published table which gives retirement age
   according to birth year */

IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN DO;
  YEARDOB = YEAR(DOBBEST) - 1;
  MONTHDOB = 12;
  DAYDOB = 31;
END;

ELSE DO;
  YEARDOB = YEAR(DOBBEST);
  MONTHDOB = MONTH(DOBBEST);
  DAYDOB = DAY(DOBBEST);
END;

BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
IF YEARDOB LE 1937 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (65*12), "SAMEDAY");
ELSE IF YEARDOB = 1938 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1939 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1940 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1941 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1942 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAMEDAY");
ELSE IF 1943 <= YEARDOB <= 1954 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (66*12), "SAMEDAY");
ELSE IF YEARDOB = 1955 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1956 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1957 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1958 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1959 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+10), "SAMEDAY");
ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
```

Appendix A.68
JCL/SAS Code: LDWSSI

```
INTNX('MONTH', BIRTHDATE, (67*12), "SAME DAY");

/* create ssi ldw indicators */

/* input arrays - psta vars are renamed for the annual files */
ARRAY PSTA(*) %DO I = 1 %TO &TOT; PST&&X&I %END;;
ARRAY T16GRS(*) %DO I = 1 %TO &TOT; T16GRSAMT&&X&I %END;;
ARRAY T16NET(*) %DO I = 1 %TO &TOT; T16NETAMT&&X&I %END;;
ARRAY T16EXL(*) %DO I = 1 %TO &TOT; T16EXLAMT&&X&I %END;;
ARRAY T16EXP(*) %DO I = 1 %TO &TOT; T16EXPAMT&&X&I %END;;
ARRAY T16PAS(*) %DO I = 1 %TO &TOT; T16PASAMT&&X&I %END;;
ARRAY EICM(*) %DO I = 1 %TO &TOT; EIN&&X&I %END;;
ARRAY UICM(*) %DO I = 1 %TO &TOT; UIN&&X&I %END;;

/* NEW ARRAYS */
ARRAY SGALEV(*) %DO I = 1 %TO &TOT; SGALEVEL&&X&I %END;;
ARRAY LDWDATE(*) %DO I = 1 %TO &TOT; LDWDATE&&X&I %END;;
ARRAY DCFEARN(*) %DO I = 1 %TO &TOT; DCFEARN&&X&I %END;;
ARRAY SSREARN(*) %DO I = 1 %TO &TOT; SSREARN&&X&I %END;;
ARRAY SSILDW(*) %DO I = 1 %TO &TOT; LDWSSI&&X&I %END;;

/* construct an array of dates corresponding to the month/year
   fields in the source arrays for LDW construction */

LENGTH LDWNAME $11;

LDWNAME = ' ';
DO I=1 TO DIM(SSILDW);
  /* construct date for the array element */
  LDWNAME = VNAME(SSILDW(I));
  YR = SUBSTR(LDWNAME, 7, 2);
  IF YR IN (94, 95, 96, 97, 98, 99) THEN YR = '19' || YR;
  ELSE YR = '20' || YR;
  YEAR = INPUT(YR, 8.);
  MONTH = INPUT(SUBSTR(LDWNAME, 9, 2), 8.);
  DAY = 1;
  LDWDATE(I) = MDY(MONTH, DAY, YEAR);
END;

/* set sga levels for time - use LDWDATEeyymm to target the
   year/month time frame. sga levels are higher for blind */

%MACRO SGAIT (DATE1, DATE2, LEV1, LEV2);
DO I = 1 TO DIM(LDWDATE);

  IF LDWDATE(I) GE &DATE1 AND LDWDATE(I) LE &DATE2 THEN DO;
    IF SUBSTR(MFT, 1, 1) = 'B' THEN SGALEV(I) = &LEV1;
    ELSE SGALEV(I) = &LEV2;
  END;

END;

%MEND;
```

Appendix A.68
JCL/SAS Code: LDWSSI

```
%SGAIT('01JAN1994'D,'31DEC1994'D,930,500);
%SGAIT('01JAN1995'D,'31DEC1995'D,940,500);
%SGAIT('01JAN1996'D,'31DEC1996'D,960,500);
%SGAIT('01JAN1997'D,'31DEC1997'D,1000,500);
%SGAIT('01JAN1998'D,'31DEC1998'D,1050,500);
/* 1999 has 2 levels */
%SGAIT('01JAN1999'D,'30JUN1999'D,1110,500);
%SGAIT('01JUL1999'D,'31DEC1999'D,1110,700);
%SGAIT('01JAN2000'D,'31DEC2000'D,1170,700);
%SGAIT('01JAN2001'D,'31DEC2001'D,1240,740);
%SGAIT('01JAN2002'D,'31DEC2002'D,1300,780);
%SGAIT('01JAN2003'D,'31DEC2003'D,1330,800);
%SGAIT('01JAN2004'D,'31DEC2004'D,1350,810);
%SGAIT('01JAN2005'D,'31DEC2005'D,1380,830);
%SGAIT('01JAN2006'D,'31DEC2006'D,1450,860);
%SGAIT('01JAN2007'D,'31DEC2007'D,1500,900);
%SGAIT('01JAN2008'D,'31DEC2008'D,1570,940);
%SGAIT('01JAN2009'D,'31DEC2009'D,1640,980);
%SGAIT('01JAN2010'D,'31DEC2010'D,1640,1000);

/* create earnings fields from eincm variables using formula.
   they will represent earnings from 1994 through march 2001 */

/* intialize ssi earnings to 0 */

DO I = 1 TO DIM(SSREARN);

   SSREARN(I) = 0;

END;

DO I = 1 TO DIM(SSREARN);

   IF EICM(I) GT 0 AND UICM(I) GT 0 THEN DO;

      SSREARN(I) = (EICM(I)*2) + 65;

   END;

   ELSE IF EICM(I) GT 0 THEN DO;

      SSREARN(I) = (EICM(I)*2) + 85;

   END;

END;

/* create earnings amounts using dcf earnings fields
   and exclusion amounts */

DO I = 1 TO DIM(DCFEARN);
/* sum gross amounts and net amounts - these are wages and
```

Appendix A.68
JCL/SAS Code: LDWSSI

```
self-employment earnings. In most cases only one or the
other will have values */
SUMEARN = SUM(OF T16GRS(I),T16NET(I),0);
/* sum the exclusion amounts */
SUMEXCEP = SUM(OF T16EXL(I),T16PAS(I),T16EXP(I),0);
/* subtract the exclusion amounts - the result are earnings
to be evaluated with reference to sga */
DCFearn(I) = SUMEARN-SUMEXCEP;

END;

/* initialize ssi ldw to 9 - reset to 0 if c01- to 9 if dead */

DO I = 1 TO DIM(PSTA);

  IF PSTA(I) GT ' ' THEN SSILDW(I) = 9;
  IF PSTA(I) = 'C01' THEN SSILDW(I) = 0;
  ELSE IF PSTA(I) = 'T01' THEN SSILDW(I) = 9;

END;

/* find first occurrence of work - psta = n01 and earnings
gt 0. up until april 2001 use the ssa eincm var.
from then on use the dcf earnings */

FOUNDWORK = 0;

DO I = 1 TO DIM(PSTA) UNTIL (FOUNDWORK GT 0);

  IF LDWDATE(I) GE '01JAN1994'D AND LDWDATE(I) LE '31MAR2001'D THEN
DO;
  IF PSTA(I) = 'N01' AND SSREARN(I) GT 0 THEN FOUNDWORK = I;
  END;
  ELSE IF PSTA(I) = 'N01' AND DCFearn(I) GT 0 THEN FOUNDWORK = I;
  END;

/* evaluate work with change in status - loop thru all the changes -
process all months indicating work -
all other codes have been set already */

IF FOUNDWORK GT 0 THEN DO;

  DO I = FOUNDWORK TO DIM(PSTA);

    IF PSTA(I) = 'N01' THEN SSILDW(I) = 1;

    ELSE IF PSTA(I) IN ('N04' 'N06') THEN DO;
    /* check if earnings are at or above sga */

    IF LDWDATE(I) GE '01JAN1994'D AND LDWDATE(I) LE '31MAR2001'D
    THEN DO;
    IF SSREARN(I) GE SGALEV(I) THEN SSILDW(I) = 1;
    END;

  END;

END;
```

Appendix A.68
JCL/SAS Code: LDWSSI

```
ELSE IF DCFEARN(I) GE SGALEV(I) THEN SSILDW(I) = 1;
END;

/* code work terminations except for death */
ELSE IF SUBSTR(PSTA(I),1,1) = 'T' AND PSTA(I) NE 'T01'
THEN SSILDW(I) = 2;

END;

END;

/* set to missing if dead or retired - pstat codes of t01 which
indicates death have been set to 9 above - overwrite to missing
using the DODBEST date from the DEMO file - the best indication
of death in the TRF. If there is not a concurrence in the t01
month
and the month of the death date then leave the LDW set to 9 */

DO I = 1 TO DIM(SSILDW);

IF (LDWDATE(I) GT RETIREDATE AND RETIREDATE GT .Z) OR
(LDWDATE(I) GT DODBEST AND DODBEST GT .Z) THEN SSILDW(I) = .;

END;

LABEL
%DO I=1 %TO &TOT;
LDWSSI&&X&I = "&&X&I SSI LEFT DUE TO WORK INDICATOR"
%END;
;
RUN;

%MEND;
%START;

PROC CONTENTS DATA=OUT.LDWSSI;
TITLE "CONTENTS OF LDWSSI";
RUN;

PROC FREQ DATA=OUT.LDWSSI;
TABLES LDWSSI:/MISSPRINT;
RUN;
```

Appendix A.69

JCL/SAS Code: LDWIND

```

//#2127LDW JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWIND) |
//*          *-----*
//* CREATE LDW INDICATOR FILE TO MERGE TO YEARLIES
//* REVISED FOR TRF08 TRF09
//* FOR DACS WE USE OPTION 2 - IF A DAC'S PRIMARY IS FLAGGED
//* ASSUME TERM AND SUSP INDICATORS ON THE DAC'S RECORD BELONG
//* TO THE PRIMARY. DO NOT FLAG THE DAC.
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* MODIFIED 12/21/2011 BY JEREMY PAGE FOR TRF10
//*****
//*
//JS010    EXEC SAS9,
//          WORK='100000,50000'
//*
//DEM      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN2
DD
DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWDATA.SA.V1,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//*
//SYSIN    DD *

/*****
*
* FILENAME: LDWIND.sas
* PROGRAMMER:MIRIAM LOEWENBERG
* PURPOSE:TO create revised DI MONTHLY LDW INDICATOR
* INPUT FILE COMBINED DEMO AND MBR DATA AND DCF DATA
* CREATED:6/29/10

*****
*/
options nocenter ls=132 ps=60 compress=binary MACROGEN MPRINT
OBS=MAX;
/* change endyr and endmn for each run */
%let begyr=1994;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

```

Appendix A.69
JCL/SAS Code: LDWIND

```
/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

/* PUT "OUT" IN BELOW!!! */
DATA TEMP.LDWDATA (KEEP=SSN CAN BIC LDW:)

  MATCHPRIM (KEEP=SSN CAN LDW:
             WORKFLAG TERMFLAG )
  MATCHSEC (KEEP=SSN CAN WORKFLAG TERMFLAG LDW:
            RENAME=(WORKFLAG=SWORKFLAG TERMFLAG=STERMFLAG))
;
MERGE DEM.DMLDW
      (KEEP=SSN CAN DOBBEST DDOBEST TWPCPLMNT1
       IN=IND)
  IN1.MBR (IN=DW KEEP=SSN BIC RFD: WIC: CDR: RFST: LAF:)
  IN2.T2WKDET1 (IN=E KEEP=COSSN TWPDATA: RENAME=(COSSN=SSN));
BY SSN;
IF IND AND DW;

/* correct bad values in twpcplmth1 by reformatting */

IF TWPCPLMNT1 GT '31DEC2011'D THEN
  TWPCPLMNT1=MDY(SUBSTR(LEFT(TWPCPLMNT1),5,2),01,
  SUBSTR(LEFT(TWPCPLMNT1),1,4));

/* use dobest from demo to compute retiredate */

/* compute retirement date for comparison */
/* if b'day is january 1st - refer to previous year */
/* sas9 offers "sameday" which simplifies the code below */
/* basis for code is published table which gives retirement age
   according to birth year */

IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN DO;
  YEARDOB = YEAR(DOBBEST) - 1;
  MONTHDOB = 12;
  DAYDOB = 31;
```

Appendix A.69

JCL/SAS Code: LDWIND

```
END;
```

```
ELSE DO;
  YEARDOB = YEAR(DOBBEST);
  MONTHDOB = MONTH(DOBBEST);
  DAYDOB = DAY(DOBBEST);
END;
```

```
BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
IF YEARDOB LE 1937 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (65*12), "SAMEDAY");
ELSE IF YEARDOB = 1938 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1939 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1940 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1941 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1942 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAMEDAY");
ELSE IF 1943 <= YEARDOB <= 1954 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (66*12), "SAMEDAY");
ELSE IF YEARDOB = 1955 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1956 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1957 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1958 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1959 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+10), "SAMEDAY");
ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (67*12), "SAMEDAY");
```

```
/* set up arrays for processing */
```

```
/* arrays for input */
ARRAY LAFI (*) %DO I = 1 %TO &TOT; LAF&&X&I %END;;
ARRAY WICI (*) %DO I = 1 %TO &TOT; WIC&&X&I %END;;
ARRAY RFDI (*) %DO I = 1 %TO &TOT; RFD&&X&I %END;;
ARRAY CDRI (*) %DO I = 1 %TO &TOT; CDR&&X&I %END;;
ARRAY RFSTI (*) %DO I = 1 %TO &TOT; RFST&&X&I %END;;
ARRAY TWPDT (*) %DO I = 1 %TO &TOT; TWPDATA&&X&I %END;;
```

```
/* for trf08 all records will be updated and rfst are on all
records - but it still may not improve the pre 2000 ldw's
appreciably */
```

```
/* begin with a check that the source values are before death
or retirement */
```

Appendix A.69
JCL/SAS Code: LDWIND

```
/* changes for trf07:
   indicators set to missing if not on the rolls that month
   twpdata helps to code work terminations */

/* changes from trf09 on:
   look for a twp completion date as starting point for
   evaluating ldw. first set the 0 for all months in pay status
   after intializing all fields to 9. this is similar to the ssi
   method.
   if the twpcmplmth is missing look for the first triplet
   indicating suspension and use it as a starting point */

ARRAY LDW1{*} %DO I=1 %TO &TOT; LDWDI&&X&I %END;;

/* find the month flagged by twpcmplmth1 to begin processing.
   flag the death or retirement index to end processing */

ARRAY LDWDATE{*} %DO I=1 %TO &TOT; LDWDATE&&X&I %END;;

LENGTH LDWNAME $9;
LDWNAME = ' ';
STOPIND = 0;
FOUNDWORK = 0;
DO I=1 TO DIM(LDW1);
/* construct date for the array element */
LDWNAME = VNAME(LDW1(I));
YR = SUBSTR(LDWNAME,6,2);
IF YR IN ('94','95','96','97','98','99') THEN YR = '19'||YR;
ELSE YR = '20'||YR;
YEAR = INPUT(YR,8.);
MONTH = INPUT(SUBSTR(LDWNAME,8,2),8.);
DAY = 1;
LDWDATE(I) = MDY(MONTH,DAY,YEAR);
END;

/* set the beginning month for processing */
DO I = 1 TO DIM(LDWDATE) UNTIL (FOUNDWORK GT 0);

IF TWPCMPLMNTH1 NE . THEN DO;
/* if twp completion before 01/94 start at the beginning */
/* twpcmplmth1 is always set to day 1 as is ldwdate */
IF TWPCMPLMNTH1 LE LDWDATE(1) THEN FOUNDWORK = 1;
ELSE IF TWPCMPLMNTH1 = LDWDATE(I) THEN FOUNDWORK = I;
END;

/* when missing twpcmplmth then use 1st suspension triplet */
ELSE IF TWPCMPLMNTH1 = . THEN DO;

/* starting at jan 1994 - look for a suspension triplet
   then start at that point when one is found */
DO I=1 TO DIM(LDWDATE);
IF RFDI(I) = '7' AND
((WICI(I) = '2' OR WICI(I) = '7'))
```

Appendix A.69
JCL/SAS Code: LDWIND

```
OR
  (RFSTI(I) IN ('EPESGA' 'DIBWRK')) THEN FOUNDWORK = I;
END; /* loop */

END; /* twpcmplmth is missing */

END; /* foundwork gt 0 */

/* set the end month for processing */
DO I = 1 TO DIM(LDWDATE) UNTIL (STOPIND GT 0);
/* build indicators only up to retirement date or death date-
missings begin in the month following the event */
IF (LDWDATE(I) GT RETIREDATE AND RETIREDATE GT .Z) OR
  (LDWDATE(I) GT DODBEST AND DODBEST GT .Z) THEN STOPIND = I;
END;

/* where no work death or retirement limits processing set the
stop index to the last month in the evaluation */
IF STOPIND = 0 THEN STOPIND = &TOT;

/* initialize the ldw variables */
DO I = 1 TO STOPIND;
IF LAFI(I) = ' ' THEN LDW1(I) = .;
ELSE IF SUBSTR(LAFI(I),1,1) IN ('C' 'E') THEN LDW1(I) = 0;
ELSE IF LAFI(I) NE ' ' THEN LDW1(I) = 9;
END;

IF FOUNDWORK GT 0 THEN DO;

DO I=FOUNDWORK TO STOPIND;
/* work suspensions */
IF RFDI(I) = '7' AND
  ((WICI(I) = '2' OR WICI(I) = '7')
  OR
  (RFSTI(I) IN ('EPESGA' 'DIBWRK')))
  THEN DO;
  LDW1(I) = 1;
  WORKFLAG = 1;
END;
/* work terminations - look at twpdatayymm in addition to cdr */
ELSE IF RFDI(I) = 'T' AND WICI(I) IN ('8' '6') AND (CDRI(I) = 'E'
OR TWPDT(I) = 'S')
  THEN DO;
  LDW1(I) = 2;
  TERMFLAG = 1;
END;
END; /*loop */

END; /* foundwork gt 0 */
DROP LDWDATE;;

/* output 3 files - 1. flagged primaries 2. flagged dacs
3. unflagged cases.
```

Appendix A.69

JCL/SAS Code: LDWIND

we do this because we need to determine whether work data on
a dac record belongs to the primary and not the dac */

```
IF WORKFLAG = 1 OR TERMFLAG = 1 THEN DO;
  IF SUBSTR(BIC,1,1) = 'C' THEN OUTPUT MATCHSEC; /* dacs */
  ELSE OUTPUT MATCHPRIM; /* primaries */
END;
ELSE OUTPUT TEMP.LDWDATA; /*no work indication on these records*/
```

```
*****This section is used if we have a merge statement issue again;
* IF WORKFLAG = 1 OR TERMFLAG = 1 THEN DO;
*   IF SUBSTR(BIC,1,1) = 'C' THEN OUTPUT MATCHSEC; /* dacs */
**  ELSE IF BIC='A' THEN OUTPUT MATCHPRIM; /* primaries */
*   ELSE OUTPUT TEMP.LDWDATA; /* not primaries or dacs */
* END;
*   ELSE OUTPUT TEMP.LDWDATA; /* not primaries or dacs */
```

RUN;

/* check the number of dac records where the ssn = can. this may be
the case for old records where the rule to always put the
dependent
boan on the record was not yet enforced. in the old days if the
bene's own ssn was not known (a dependent bene) the ssn of the
primary was entered into the boan field */

```
PROC FREQ DATA=TEMP.LDWDATA;
  TABLES BIC; WHERE SUBSTR(BIC,1,1) = 'C' AND CAN = SSN;
  TITLE 'CAN = SSN AND BIC = C';
RUN;
```

```
PROC SORT DATA=MATCHPRIM; BY CAN; RUN;
PROC FREQ; TABLES WORKFLAG*TERMFLAG/LIST MISSING;
  TITLE 'PRIMARY CASES WITH WORK INDICATED';
RUN;
```

```
PROC SORT DATA=MATCHSEC; BY CAN; RUN;
PROC FREQ; TABLES SWORKFLAG*STERMFLAG/LIST MISSING;
  TITLE 'DAC CASES WITH WORK INDICATED';
RUN;
```

```
DATA MATCHSEC (KEEP=SSN LDW: );
MERGE MATCHPRIM (IN=P DROP=SSN LDW:)
  MATCHSEC (IN=S); BY CAN;
```

/* keep only dac cases */

```
ARRAY LDW{*} %DO I=1 %TO &TOT; LDWDI&&X&I %END;;
```

/* if dac has a primary who has been flagged - reset indicator
to 9 for the dac because the work indication most likely
belongs to the primary */

Appendix A.69
JCL/SAS Code: LDWIND

```
IF P AND S AND CAN NE ' ' THEN DO;
  IF WORKFLAG = 1 AND SWORKFLAG = 1 THEN DO;
    DO I = 1 TO DIM(LDW);
      IF LDW(I) = 1 THEN LDW(I) = 9;
    END;
  END;
  IF TERMFLAG = 1 AND STERMFLAG = 1 THEN DO;
    DO I = 1 TO DIM(LDW);
      IF LDW(I) = 2 THEN LDW(I) = 9;
    END;
  END;
END;

IF S;

RUN;

PROC PRINT DATA=MATCHSEC (OBS=25); TITLE 'REVISED FLAGGED DAC
RECORDS';
RUN;
PROC PRINT DATA=MATCHPRIM (OBS=25); TITLE 'FLAGGED PRIMARY RECORDS';
RUN;

/* put the pieces back together */
PROC SORT DATA= MATCHPRIM; BY SSN; RUN;
PROC SORT DATA= MATCHSEC; BY SSN; RUN;
DATA TEMP1.LDWDATA (KEEP=SSN LDW: );
SET TEMP.LDWDATA MATCHPRIM MATCHSEC;
BY SSN;
RUN;
/* dedup the few duplicates because of bad boans */
PROC SORT DATA=TEMP1.LDWDATA OUT=OUT.LDWDATA NODUPKEY; BY SSN; RUN;
PROC FREQ DATA=OUT.LDWDATA;
TABLES LDW:/ MISSPRINT;
RUN;
PROC PRINT DATA=OUT.LDWDATA (OBS=25); RUN;

%MEND;

%START;
```

Appendix A.70
JCL/SAS Code: LDWCOMB

```
//#2127LDC JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWCOMB) |
//*          *-----*
//*
//* CREATE REVISED COMBINED LDW INDICATORS FOR TRF10
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* CONSTRUCT COMBINED LEFT DUE TO WORK INDICATORS
//* MODIFIED 12/23/2011 BY JEREMY PAGE
//*****
//*
//JS010 EXEC SAS9,
//      WORK='120000,60000'
//*
//TEMP1 DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      SPACE=(CYL,(1000,100),RLSE),VOL=(,,8)
//IN1   DD DSN=OPDR.TG.PRD.ETTW.#2127.LDWSSI.Y2010.SSD,DISP=SHR
//IN2   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWDATA.SA.V1,DISP=SHR
//IN3   DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//FIN   DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWCOMB.SA.V1,
//      DISP=(OLD,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY OBS=MAX;

%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);
```

Appendix A.70
JCL/SAS Code: LDWCOMB

```
DATA OUT.LDWCOMB (KEEP=SSN LDWSSI: LDWDI: LDWCM:);
MERGE IN1.LDWSSI
      ( KEEP=SSN
        %DO I=1 %TO &TOT;
          LDWSSI&&X&I
        %END; )
      IN2.LDWDATA
      ( KEEP=SSN
        %DO I=1 %TO &TOT;
          LDWDI&&X&I
        %END; )
      IN3.MBR
      ( KEEP=SSN
        %DO I=1 %TO &TOT;
          RFST&&X&I
        %END; )
      FIN.LINKSSN
      ( KEEP=SSN IN=L9)
      ;
BY SSN;
IF L9;

/* new 2008 - create a program-combined ldw indicator */

ARRAY NEWLDW (*) %DO I=1 %TO &TOT; LDWCM&&X&I %END;;
ARRAY DILDW  (*) %DO I=1 %TO &TOT; LDWDI&&X&I %END;;
ARRAY SSILDW (*) %DO I=1 %TO &TOT; LDWSSI&&X&I %END;;
ARRAY RFST   (*) %DO I=1 %TO &TOT; RFST&&X&I %END;;

DO I = 1 TO DIM(NEWLDW);
  /* if not in one pgm combined ldw=ldw from other pgm */
  IF SSILDW(I)=. THEN NEWLDW(I)=DILDW(I);
  ELSE IF DILDW(I)=. THEN NEWLDW(I)=SSILDW(I);
  /* if in current pay status in either pgm combined ldw=0 */
  ELSE IF DILDW(I)=0 OR SSILDW(I)=0 THEN NEWLDW(I)=0;
  /* new for trf09 check prison suspension on di side */
  /* if suspended due to work in either pgm combined ldw=1 */
  ELSE IF DILDW(I)=1 OR (SSILDW(I)=1 AND RFST(I) NE 'PRISON')
    THEN NEWLDW(I) = 1;
  /* if terminated due to work in either pgm combine ldw=2 */
  ELSE IF DILDW(I)=2 OR (SSILDW(I)=2 AND RFST(I) NE 'PRISON')
    THEN NEWLDW(I) = 2;
  /* if inelig for other reason in both pgms combine ldw=9 */
  ELSE NEWLDW(I)=9;
END;

RUN;
%MEND;
%START;

PROC CONTENTS DATA=OUT.LDWCOMB;
TITLE "CONTENTS OF LDWCOMB";
RUN;
```

Appendix A.70
JCL/SAS Code: LDWCOMB

```
PROC PRINT DATA=OUT.LDWCOMB (OBS=25); RUN;  
PROC FREQ DATA=OUT.LDWCOMB;  
  TABLES LDW: / MISSING;  
RUN;
```

Appendix A.71
JCL/SAS Code: LDWSTRNG

```
// #2127STR JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
// *****
// *      *-----*
// *      | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWSTRNG) |
// *      *-----*
// *
// * PURPOSE: STRING OUT 3'S IN THE LDW INDICATORS
// * CONTACT JEREMY PAGE
// * SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
// * E-MAIL JPAGE@MATHEMATICA-MPR.COM
// *
// * CREATED: 2/2/2011 BY JEREMY PAGE
// *****
// *
// JS010    EXEC SAS9,
//          WORK='120000,60000'
// *
// IN       DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWCOMB.SA.V1,DISP=SHR
// DEM      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
// OUT      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
// SYSIN    DD *
OPTIONS NOCENTER MACROGEN COMPRESS=BINARY OBS=MAX;

%MACRO DOIT;

DATA OUT.LDWSTRNG (KEEP=SSN LDWSSI: LDWDI: LDWCM:);
  MERGE IN.LDWCOMB
        DEM.DMLDW (KEEP=SSN DOBBEST DODBEST);
  BY SSN;

***CALCULATE FRA;
***WHEN CALCULATING FRA PEOPLE BORN ON JAN 1 ARE CONSIDERED
***TO HAVE A BIRTH DATE OF DEC 31 OF THE PREVIOUS YEAR.;
IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN
  RETIRE_DOB=DOBBEST-1;
ELSE RETIRE_DOB=DOBBEST;

  IF YEAR(RETIRE_DOB) LE 1937 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,(65*12),"SAME DAY");
  ELSE IF YEAR(RETIRE_DOB) = 1938 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+2),"SAME DAY");
  ELSE IF YEAR(RETIRE_DOB) = 1939 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+4),"SAME DAY");
  ELSE IF YEAR(RETIRE_DOB) = 1940 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+6),"SAME DAY");
  ELSE IF YEAR(RETIRE_DOB) = 1941 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+8),"SAME DAY");
  ELSE IF YEAR(RETIRE_DOB) = 1942 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+10),"SAME DAY");
  ELSE IF 1943 <= YEAR(RETIRE_DOB) <= 1954 THEN RETIREDATE =
```

Appendix A.71
JCL/SAS Code: LDWSTRNG

```
INTNX('MONTH',RETIRE_DOB,(66*12),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1955 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,((66*12)+2),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1956 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,((66*12)+4),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1957 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,((66*12)+6),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1958 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,((66*12)+8),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1959 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,((66*12)+10),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) GE 1960 THEN RETIREDATE =
INTNX('MONTH',RETIRE_DOB,(67*12),"SAMEDAY");

*****
***CONSTRUCT A DATE THAT IS THE EARLIEST OF FRA
***DOD OR DEC 31, 2010. THIS WILL ALLOW THE LOOP TO
***TO STOP STRINGING 3 WHEN A PERSON IS DEAD OR FRA.
*****;
ENDDATE=MIN(RETIREDATE,DODBEST,'31DEC2010'D);

STOPMNTN=INTCK('MONTH','01JAN1994'D,ENDDATE)+1;

***ARRAY REFERENCES FOR LDW VARIABLES;
ARRAY SSI_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWSSI&YR.01-LDWSSI&YR.12
                %END;;

ARRAY DI_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWDI&YR.01-LDWDI&YR.12
                %END;;

ARRAY COMB_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWCM&YR.01-LDWCM&YR.12
                %END;;

***STRING 3 IN LDW VARIABLES;
DO I = 2 TO STOPMNTN;
  ***STRING LDWSSI;
  IF SSI_ARRAY(I) = . AND SSI_ARRAY(I-1) IN (2,3)
    THEN SSI_ARRAY(1) = 3;

  ***STRING LDWDI;
  IF DI_ARRAY(I) = . AND DI_ARRAY(I-1) IN (2,3)
    THEN DI_ARRAY(1) = 3;

  ***STRING LDWCOMB;
  IF COMB_ARRAY(I) = . AND COMB_ARRAY(I-1) IN (2,3)
    THEN COMB_ARRAY(1) = 3;
END;
```

Appendix A.71
JCL/SAS Code: LDWSTRNG

```
PROC CONTENTS DATA=OUT.LDWSTRNG;  
RUN;
```

```
%MEND DOIT;  
%DOIT;
```

Appendix A.72
JCL: Y9495JCL

```
// #212794 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,  
//           MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127  
// *****  
// *  
// *           *-----*  
// *           |OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(Y9495JCL)|  
// *           *-----*  
// *  
// * CREATE YEARLY FILES FOR 1994-1995  
// * MONTHLY FIELDS FROM JAN 1994-DEC 1995  
// * DCF VARIABLES NOT INCLUDED BEFORE 2000  
// * CONTACT JEREMY PAGE  
// * SSA PHONE 202 358-6228 MPR PHONE 202 554-7515  
// * E-MAIL JPAGE@MATHEMATICA-MPR.COM  
// * CREATED BY: MIRIAM LOEWENBERG  
// * MODIFIED BY: JEREMY PAGE 3/1/12  
// *  
// *NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.  
// * PLEASE DO NOT CANCEL IT.  
// *  
// *****  
// JS010 EXEC SAS9,  
//        WORK='100000,50000',  
//        PARM='SYSPARM="1994" '  
// *  
// OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1994.SA.V2,DISP=SHR  
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR  
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR  
// IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR  
// IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR  
// IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR  
// IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR  
// FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR  
// OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1994.SA.V1,  
//        DISP=(NEW,CATLG,DELETE),  
//        UNIT=TSILO  
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),  
//        DISP=(SHR,PASS,KEEP)  
// *  
// JS020 EXEC SAS9,  
//        WORK='100000,50000',  
//        PARM='SYSPARM="1995" '  
// *  
// OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1995.SA.V2,DISP=SHR  
// IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR  
// IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR  
// IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR  
// IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR  
// IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR  
// IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR  
// FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR  
// OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1995.SA.V1,  
//        DISP=(NEW,CATLG,DELETE),
```

Appendix A.72
JCL: Y9495JCL

```
//          UNIT=TSILO  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),  
//          DISP=(SHR,PASS,KEEP)  
//*
```

Appendix A.73
JCL: Y9697JCL

```
//#212796 JOB (12510000,T715,,SAS,,ITC9FL),2121PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*      *-----*
//*      |OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(Y9697JCL)|
//*      *-----*
//* CREATE YEARLY FILES FOR 1996-1997
//* MONTHLY FIELDS FROM JAN 1996-DEC 1997
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CONTACT JEREMY PAGE
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 3/1/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="1996"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1996.SA.V2,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1996.SA.V1,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),
//      DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="1997"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1997.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1997.SA.V1,
```

Appendix A.73
JCL: Y9697JCL

```
//          DISP=(NEW,CATLG,DELETE),  
//          UNIT=TSILO  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),  
//          DISP=(SHR,PASS,KEEP)
```

Appendix A.74
JCL: Y9899JCL

```
//#212798 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//                MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(Y9899JCL)|
//*          *-----*
//* CREATE YEARLY FILES FOR 1996-1997
//* MONTHLY FIELDS FROM JAN 1996-DEC 1997
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CONTACT JEREMY PAGE
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 3/1/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010   EXEC SAS9,
//        WORK='100000,50000',
//        PARM='SYSPARM="1998"'
//*
//OLD    DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1998.SA.V1,DISP=SHR
//IN1    DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2    DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3    DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//FIN    DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1998.SA.V1,
//        DISP=(NEW,CATLG,DELETE),
//        UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),
//        DISP=(SHR,PASS,KEEP)
//*
//JS020   EXEC SAS9,
//        WORK='100000,50000',
//        PARM='SYSPARM="1999"'
//*
//OLD    DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y1999.SA.V1,DISP=SHR
//IN1    DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2    DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3    DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6    DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//FIN    DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1999.SA.V1,
```

Appendix A.74
JCL: Y9899JCL

```
//          DISP=(NEW,CATLG,DELETE),  
//          UNIT=TSILO  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR1),  
//          DISP=(SHR,PASS,KEEP)  
//*
```

Appendix A.75
SAS Code: T2T16YR1

```

/*****
*
* FILENAME: T2T16YR1
* PROGRAMMER: JEREMY PAGE
* PURPOSE: TO CREATE YEARLY FILE FOR TRF10
* CREATED: 3/1/12
*****/

*/
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY;
/* change endyr and endmmn for each run */
%LET YR = SUBSTR(&SYSPARM,3,2);
/* CHANGE ENDYR AND ENDMMN FOR EACH RUN */

%let begyr=&SYSPARM;
%let endyr=&SYSPARM; /* change as needed */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */
%macro start;
%let k=1;
%do i=&begyr %to &endyr;
%if &i<2000 %then %let yr=%eval(&i-1900);
%else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
%else %let yr=%eval(&i-2000);
%if &i=&endyr %then %let emn=%eval(&endmn);
%else %let emn=12;
%do j=1 %to &emn;
%if &j<10 %then %let mn=0%eval(&j);
%else %let mn=%eval(&j);
%let x&k=%eval(&yr)%eval(&mn);
%let k=%eval(&k+1);
%end;
%end;
%let tot=%eval(&k-1);

DATA OUT1.Y&SYSPARM;
MERGE
OLD.Y&SYSPARM
FIN.LINKSSN (IN=INFIN)
IN1.LONG(IN = INL KEEP=SSN
%DO I=1 %TO &TOT;
EIN&&X&I FDA&&X&I
FDP&&X&I STP&&X&I
SPA&&X&I UIN&&X&I
LVF&&X&I MDT&&X&I
PST&&X&I STC&&X&I
T16DX1&&X&I
T16DX2&&X&I
%END;
RENAME=(
%DO I=1 %TO &TOT;

```

Appendix A.75
SAS Code: T2T16YR1

```

      EIN&&X&I=EICM&&X&I  FDA&&X&I=FAMT&&X&I
      FDP&&X&I=FPMT&&X&I  STP&&X&I=SPMT&&X&I
      SPA&&X&I=SAMT&&X&I  UIN&&X&I=UINC&&X&I
      LVF&&X&I=LIVF&&X&I  MDT&&X&I=MTST&&X&I
      PST&&X&I=PSTA&&X&I  STC&&X&I=SCON&&X&I
    %END; ) )
IN2.MBR ( IN=INM KEEP=SSN HI_START HI_TERM SMI_STAR SMI_TERM
      %DO I=1 %TO &TOT; MBC&&X&I LAF&&X&I WIC&&X&I RFD&&X&I
      RFST&&X&I  BPD&&X&I DPEN&&X&I DUEO&&X&I PAYD&&X&I PAYO&&X&I
      T2DX1&&X&I T2DX2&&X&I MBA&&X&I MBP&&X&I CDR&&X&I
      DIRPAY&&X&I MEDPREM&&X&I
      %END;
      RENAME=(
      %DO I=1 %TO &TOT; MBC&&X&I=DUED&&X&I
      %END; ) )
IN3.ALLCDR
      ( IN=ALL KEEP=SSN
      %DO I=1 %TO &TOT;
      ALX&&X&I
      EDX&&X&I
      MIEX&&X&I
      %END; )
IN4.LDWSTRNG
      ( KEEP=SSN
      %DO I=1 %TO &TOT;
      LDWDI&&X&I
      LDWSSI&&X&I
      LDWCM&&X&I
      %END; )
IN5.T2EARN
      ( KEEP=COSSN
      %DO I=1 %TO &TOT;
      T2GRSAMT&&X&I
      T2NETAMT&&X&I
      T2SEHRS&&X&I
      T2SEVERIND&&X&I
      T2VERIND&&X&I
      %END;
      RENAME=( COSSN=SSN ) )
IN6.T2WKDETN
      ( KEEP=COSSN
      %DO I=1 %TO &TOT;
      TWPDATA&&X&I
      T2CDNAMT&&X&I
      T2EXPAMT&&X&I
      T2FRAUDVER&&X&I
      T2SBDYAMT&&X&I
      T2UBEAMT&&X&I
      %END;
      RENAME=( COSSN=SSN ) )
;
BY SSN;
```

Appendix A.75
SAS Code: T2T16YR1

```
/* DELETE CASES NOT IN THE MASTER FINDER */

IF INFIN ;

/* INPUT AFFAYS */
ARRAY FPMT(*) %DO I=1 %TO &TOT; FPMT&&X&I %END;;
ARRAY FAMT(*) %DO I=1 %TO &TOT; FAMT&&X&I %END;;
ARRAY PSTA(*) %DO I=1 %TO &TOT; PSTA&&X&I %END;;
ARRAY MTST(*) %DO I=1 %TO &TOT; MTST&&X&I %END;;
ARRAY SPMT(*) %DO I=1 %TO &TOT; SPMT&&X&I %END;;
ARRAY SAMT(*) %DO I=1 %TO &TOT; SAMT&&X&I %END;;
ARRAY SCON(*) %DO I=1 %TO &TOT; SCON&&X&I %END;;
ARRAY DBEN(*) %DO I=1 %TO &TOT; DUED&&X&I %END;;
ARRAY UICM(*) %DO I=1 %TO &TOT; UINC&&X&I %END;;
ARRAY EICM(*) %DO I=1 %TO &TOT; EICM&&X&I %END;;
ARRAY LAFCODE(*) %DO I=1 %TO &TOT; LAF&&X&I %END;;
ARRAY LDWD(*) %DO I=1 %TO &TOT; LDWDI&&X&I %END;;

ARRAY PROA(*) %DO I=1 %TO &TOT; PROA&&X&I %END;;
ARRAY PROB(*) %DO I=1 %TO &TOT; PROB&&X&I %END;;
ARRAY SBEN(*) %DO I=1 %TO &TOT; PAYS&&X&I %END;;
ARRAY SDUE(*) %DO I=1 %TO &TOT; DUES&&X&I %END;;

/* REVISION 2008 -NO INITIALIZATION TO 0 FOR ALL CASES - VARS WILL
BE MISSING IF NOT IN SSI - THIS IS NECESSARY BECAUSE IN PAST
VERSIONS OF TRF - SSI AMOUNT VARS WERE INTIALIZED TO 0 FOR ALL
CASES - RECORDS THAT ARE NOT OVERWRITTEN BY NEW FINDER PULLS
RETAIN THE 0'S */

/* REVISION 2009 - REMOVE THE 0 FROM THE CODE */
DO I = 1 TO DIM(FPMT);
  SBEN(I) = SUM(OF FPMT(I),SPMT(I));
  SDUE(I) = SUM(OF FAMT(I),SAMT(I));
END;

/* CONCURRENT BENEFICIARY INDICATOR */
/* REVISION 2008 - INDICATOR WILL FLAG BENEFICIARIES RECEIVING
A PAYMENT (IN CURRENT PAY) IN BOTH PROGRAMS, SSI AND SSDI */
ARRAY CONC(*) %DO I=1 %TO &TOT; CONC&&X&I %END;;

DO I = 1 TO DIM(DBEN);
  IF (SDUE(I) > 0 AND PSTA(I) IN ('C01' 'M01' 'M02')) AND
  (DBEN(I) > 0 AND SUBSTR(LAFCODE(I),1,1) IN ('C' 'E'))
  THEN CONC(I) = 1;
  ELSE CONC(I) = 0;
END;

/* CREATE TERMINATION INDICATOR */
ARRAY TERMA(*) %DO I=1 %TO &TOT; TSSI&&X&I %END;;
ARRAY TERMB(*) %DO I=1 %TO &TOT; TSSD&&X&I %END;;

/* CREATE TERMINATION INDICATOR FOR SSI */
```

Appendix A.75

SAS Code: T2T16YR1

```

/* TAKE THE INITIALIZATION TO 0 OUT OF THE CODE - 2009 */
DO I = 1 TO DIM(PSTA);
  IF SUBSTR(PSTA(I),1,1) = 'T' THEN TERMA(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN TERMA(I) = 0;
  IF TERMA(I)=1 THEN TERMSI=1; /* WILL BE WRITTEN FOR EACH YEAR */
END;

/* CREATE TEMINATION INDICATOR FOR SSDI */

DO I = 1 TO DIM(LAFCODE);

IF SUBSTR(LAFCODE(I),1,1) IN ('T','X') THEN TERMB(I)=1;
ELSE IF LAFCODE(I) NE ' ' THEN TERMB(I)=0;
  IF TERMB(I)=1 THEN TERMSSD=1;
END;

/* CREATE MONTHLY INDICATORS FOR 1619A-1619B*/

/* PROB */

DO I = 1 TO DIM(MTST);
  IF MTST(I) IN ('A','B','F') AND
  PSTA(I) IN ('N01', 'P01', 'E01') AND
  EICM(I) > 0
  THEN PROB(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN PROB(I) = 0;
END;
/* PROA */
DO I = 1 TO DIM(SCON);
/*FOR PROA (THIS SHOULD YIELD NO CASES WHERE PROB = 1 AND PROA =
1)*/
  IF SCON(I) IN ('D','E','F','G') AND
  PSTA(I) IN ('C01', 'M01', 'M02') AND
  SDUE(I) > 0 AND
  EICM(I) > 0
  THEN PROA(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN PROA(I) = 0;
END;

/* CREATE A MEDICARE INDICATOR */

ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;
ARRAY MEDR (*) %DO I=1 %TO &TOT; MEDR&&X&I %END;;
J=1;
Y=&BEGYR;
DO I = 1 TO DIM(CYEAR);
  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
  IF INT(I/12)=I/12 THEN DO;
    J=1;
    Y=Y+1;
  END;
  ELSE J=J+1;

```

Appendix A.75
SAS Code: T2T16YR1

```

NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4)||SUBSTR(CYEAR(I),5,2),YYMMN6.);

MEDR(I) = 0;
IF (HI_START > .Z AND HI_START <= NYEAR(I)) OR
   (SMI_STAR > .Z AND SMI_STAR <= NYEAR(I) <= SMI_TERM)
   THEN MEDR(I) = 1;
ELSE IF (HI_START > .Z AND (NYEAR(I) <= HI_START OR (HI_TERM > .Z
AND
   NYEAR(I) >= HI_TERM))) OR (SMI_STAR > .Z AND (NYEAR(I) <=
SMI_STAR
   OR (SMI_TERM > .Z AND NYEAR(I) >= SMI_TERM))) THEN MEDR(I) = 0;
ELSE MEDR(I) = .;

END;

DROP Y J I %DO I=1 %TO &TOT; C&&X&I N&&X&I %END;;
DROP HI_START HI_TERM SMI_STAR SMI_TERM;

ARRAY LONE (*) %DO I=1 %TO &TOT; LONE&&X&I %END;;
ARRAY OTHR (*) %DO I=1 %TO &TOT; OTHR&&X&I %END;;
ARRAY FACL (*) %DO I=1 %TO &TOT; FACL&&X&I %END;;
ARRAY LIVF (*) %DO I=1 %TO &TOT; LIVF&&X&I %END;;

DO I = 1 TO DIM(LIVF);
  IF LIVF(I) = 'A' THEN LONE(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN LONE(I) = 0;
  ELSE LONE(I) = .;

  IF LIVF(I) = 'B' THEN OTHR(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN OTHR(I) = 0;
  ELSE OTHR(I) = .;

  IF LIVF(I) = 'D' THEN FACL(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN FACL(I) = 0;
  ELSE FACL(I) = .;
END;

LENGTH
%DO I=1 %TO &TOT;
MBA&&X&I
MBP&&X&I
DUED&&X&I
PAYD&&X&I
PAYO&&X&I
DUEO&&X&I
FAMT&&X&I
FPMT&&X&I
SAMT&&X&I
PAYS&&X&I
DUES&&X&I
SPMT&&X&I
UINC&&X&I

```

Appendix A.75
SAS Code: T2T16YR1

```
TSSI&&X&I
TSSD&&X&I
CONC&&X&I
OTHR&&X&I
PROA&&X&I
LONE&&X&I
FACL&&X&I
MEDR&&X&I
DIRPAY&&X&I
MEDPREM&&X&I
%END;
```

4

```
TERMSSD TERMSSI 3
;
```

LABEL

```
TERMSSI = 'TERMINATION INDICATOR T16 FOR YEAR'
TERMSSD = 'TERMINATION INDICATOR T2 FOR YEAR'
%DO I=1 %TO &TOT;
DUEO&&X&I = "&&X&I DEPENDENT PAYMENT DUE"
DPEN&&X&I = "&&X&I NUMBER OF DEPENDENTS"
EICM&&X&I = "&&X&I COUNTABLE EARNED INCOME"
CONC&&X&I = "&&X&I CONCURRENTLY IN CURRENT PAY"
DUED&&X&I = "&&X&I FEDERAL SSDI BENEFIT CREDITED"
PAYD&&X&I = "&&X&I PHUS SSDI BENEFIT PAID"
PAYO&&X&I = "&&X&I PHUS SSDI DEPENDENT BENEFIT PAID"
MBA&&X&I = "&&X&I FEDERAL SSDI BENEFIT DUE"
MBP&&X&I = "&&X&I FEDERAL SSDI BENEFIT PAID"
FACL&&X&I = "&&X&I LIVES IN MEDICAL FACILITY"
FAMT&&X&I = "&&X&I FEDERAL SSI BENEFIT DUE"
FPMT&&X&I = "&&X&I FEDERAL SSI BENEFIT PAID"
LIVF&&X&I = "&&X&I LIVING ARRANGMENT"
LONE&&X&I = "&&X&I LIVES ALONE"
OTHR&&X&I = "&&X&I LIVES W/ ANOTHER SSI RECIPIENT"
PROA&&X&I = "&&X&I CONTINUING SSI ELIG - 1619(A)"
PSTA&&X&I = "&&X&I PAYMENT STATUS"
MTST&&X&I = "&&X&I INCOME TEST"
SAMT&&X&I = "&&X&I STATE SSI BENEFIT DUE"
PAYS&&X&I = "&&X&I SSI BENEFIT PAID"
SCON&&X&I = "&&X&I STATE CONCURRENT ELIGIBILITY IND"
DUES&&X&I = "&&X&I SSI BENEFIT DUE"
SPMT&&X&I = "&&X&I STATE SSI BENEFIT PAID"
UINC&&X&I = "&&X&I UNEARNED INCOME"
LAF&&X&I = "&&X&I LEDGER ACCOUNT FILE STATUS"
TSSI&&X&I = "&&X&I MONTHLY TERMINATED STATUS-T16"
TSSD&&X&I = "&&X&I MONTHLY TERMINATED STATUS-T2"
MEDR&&X&I = "&&X&I MEDICARE INDICATOR-T2"
DIRPAY&&X&I = "&&X&I PHUS DIRECT PAY"
MEDPREM&&X&I = "&&X&I PHUS MEDICARE PREMIUM"
LDWSSI&&X&I = "&&X&I SSI LEFT DUE TO WORK INDICATOR"
LDWDI&&X&I = "&&X&I SSDI LEFT DUE TO WORK INDICATOR"
LDWCM&&X&I = "&&X&I COMBINED LEFT DUE TO WORK INDICATOR"
%END;
```

Appendix A.75
SAS Code: T2T16YR1

```
SSN      = 'SOCIAL SECURITY NUMBER'
;

/* CREATE DIAGNOSIS VARIABLES */

ARRAY T2D1(*) %DO I = 1 %TO &TOT; T2DX1&&X&I %END;;
ARRAY T2D2(*) %DO I = 1 %TO &TOT; T2DX2&&X&I %END;;
ARRAY T16D1(*) %DO I = 1 %TO &TOT; T16DX1&&X&I %END;;
ARRAY T16D2(*) %DO I = 1 %TO &TOT; T16DX2&&X&I %END;;
ARRAY CDX1(*) $4 %DO I = 1 %TO &TOT; DX1X&&X&I %END;;
ARRAY CDX2(*) $4 %DO I = 1 %TO &TOT; DX2X&&X&I %END;;

/* COLLAPSE THE DIAGNOSIS CODES TO 1 CODE -
T2 TAKES PRECEDENCE IF BOTH ARE PRESENT */

/* BOTH T2 AND CONCS */
DO I = 1 TO DIM(T2D1);
  CDX1(I) = T2D1(I);
  CDX2(I) = T2D2(I);
END;

/* T16 ONLIES */
DO I = 1 TO DIM(T2D1);
  IF CDX1(I) = ' ' THEN CDX1(I) = T16D1(I);
  IF CDX2(I) = ' ' THEN CDX2(I) = T16D2(I);
END;

DROP %DO I = 1 %TO &TOT; T2DX1&&X&I T16DX1&&X&I
      T2DX2&&X&I T16DX2&&X&I %END;;
DROP I ;

LABEL
%DO I=1 %TO &TOT;
PROB&&X&I = "&&X&I CONTINUING MEDICAID ELIG - 1619(B)"
%END;
;
DROP I;
RUN;
%MEND;
%START;

proc contents DATA=OUT1.Y&SYSPARM;
title "CONTENTS OF YEAR &SYSPARM";
run;

PROC FREQ DATA=OUT1.Y&SYSPARM; TABLES LAF: PSTA: LDWDI: LDWSSI:
      LDWCM:;
RUN;
PROC MEANS DATA=OUT1.Y&SYSPARM; VAR
DUED:
PAYD:
PAYO:
```

Appendix A.75
SAS Code: T2T16YR1

```
DUEO:  
FAMT:  
FPMT:  
SAMT:  
PAYS:  
DUES:  
SPMT:  
UINC:  
EICM:  
PROA:  
PROB:  
DIRPAY:  
MEDPREM:  
;  
RUN;  
PROC PRINT DATA=OUT1.Y&SYSPARM(OBS=25); RUN;
```

Appendix A.76
JCL: Y0100JCL

```
//#2127Y01 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          |  OPDR.TG.PRD.ETTW.N8043.LIB10(Y0100JCL)  |
//*          *-----*
//* CREATE YEARLY FILES FOR 2001-2000
//* MONTHLY FIELDS FROM JAN 2001-DEC 2000
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 2/9/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010    EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2001"'
//*
//OLD      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2001.SA.V1,DISP=SHR
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//IN7      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT1     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2001.SA.V1,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//          DISP=(SHR,PASS,KEEP)
//*
//JS020    EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2000"'
//*
//OLD      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2000.SA.V1,DISP=SHR
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4      DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
```

Appendix A.76
JCL: Y0100JCL

```
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2000.SA.V1,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.77

JCL: Y0302JCL

```
//#2127Y03 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//                               MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*   *-----*
//*   | OPDR.TG.PRD.ETTW.N8043.LIB10(Y0302JCL) |
//*   *-----*
//* CREATE YEARLY FILES FOR 2003-2002
//* MONTHLY FIELDS FROM JAN 2003-DEC 2002
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 2/9/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010   EXEC SAS9,
//         WORK='100000,50000',
//         PARM='SYSPARM="2003"'
//*
//OLD     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2003.SA.V1,DISP=SHR
//IN1     DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//IN7     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN     DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//         DISP=(NEW,DELETE,DELETE),
//         UNIT=TSILO
//OUT1    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2003.SA.V1,
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=TSILO
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//         DISP=(SHR,PASS,KEEP)
//*
//JS020   EXEC SAS9,
//         WORK='100000,50000',
//         PARM='SYSPARM="2002"'
//*
//OLD     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2002.SA.V1,DISP=SHR
//IN1     DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
```

Appendix A.77
JCL: Y0302JCL

```
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2002.SA.V1,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//      DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.78

JCL: Y0604JCL

```

//#2127Y06 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//                MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB10(Y0604JCL) |
//*          *-----*
//* CREATE YEARLY FILES FOR 2006-2004
//* MONTHLY FIELDS FROM JAN 2006-DEC 2004
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 2/9/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010 EXEC SAS9,
//        WORK='100000,50000',
//        PARM='SYSPARM="2006"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2006.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//TEMP DD DSN=&&TEMP,
//        DISP=(NEW,DELETE,DELETE),
//        UNIT=TSILO
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2006.SA.V1,
//        DISP=(NEW,CATLG,DELETE),
//        UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//        DISP=(SHR,PASS,KEEP)
//*
//JS020 EXEC SAS9,
//        WORK='100000,50000',
//        PARM='SYSPARM="2005"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2005.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR

```

Appendix A.78
JCL: Y0604JCL

```
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2005.SA.V1,
// DISP=(NEW,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
// DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
// WORK='100000,50000',
// PARM='SYSPARM="2004"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2004.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2004.SA.V1,
// DISP=(NEW,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
// DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.79
JCL: Y0907JCL

```
//#2127Y09 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N8043.LIB10(Y0907JCL) |
//*          *-----*
//*
//* CREATE YEARLY FILES FOR 2009-2007
//* MONTHLY FIELDS FROM JAN 2009-DEC 2007
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CREATED BY: MIRIAM LOEWENBERG
//* MODIFIED BY: JEREMY PAGE 2/9/12
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//*NOTE: THIS JOB IS EXPECTED TO RUN FOR AT LEAST 6 DAYS.
//*      PLEASE DO NOT CANCEL IT.
//*
//*****
//*
//JS010     EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2009"'
//*
//OLD      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2009.SA.V1,DISP=SHR
//IN1     DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//IN7     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN     DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT1    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2009.SA.V1,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN   DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
//          DISP=(SHR,PASS,KEEP)
//*
//JS020     EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2008"'
//*
//OLD      DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2008.SA.V1,DISP=SHR
//IN1     DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
```

Appendix A.79
JCL: Y0907JCL

```
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2008.SA.V1,
// DISP=(OLD,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
// DISP=(SHR,PASS,KEEP)
//*
//JS030 EXEC SAS9,
// WORK='100000,50000',
// PARM='SYSPARM="2007"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF09P.Y2007.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//IN7 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8 DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT1 DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2007.SA.V1,
// DISP=(NEW,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16YR2),
// DISP=(SHR,PASS,KEEP)
//*
```

Appendix A.80
SAS Code: T2T16YR2

```
/* *****  
*  
* FILENAME: T2T16YR2  
* PURPOSE: TO CREATE YEARLY FILE FOR TRF10  
* CREATED: 7/13/06 BY MIRIAM LOEWENBERG  
* MODIFIED: 2/9/10 BY JEREMY PAGE  
  
*****  
*/  
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY OBS=MAX;  
/* change endyr and endmmn for each run */  
  
%LET YR = SUBSTR(&SYSPARM,3,2);  
/* CHANGE ENDYR AND ENDMMN FOR EACH RUN */  
  
%let begyr=&SYSPARM;  
%let endyr=&SYSPARM; /* change as needed */  
%let endmn=12; /* change as needed */  
  
/* step to assign macro variables to handle time series data */  
%macro start;  
%let k=1;  
%do i=&begyr %to &endyr;  
%if &i<2000 %then %let yr=%eval(&i-1900);  
%else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);  
%else %let yr=%eval(&i-2000);  
%if &i=&endyr %then %let emn=%eval(&endmn);  
%else %let emn=12;  
%do j=1 %to &emn;  
%if &j<10 %then %let mn=0%eval(&j);  
%else %let mn=%eval(&j);  
%let x&k=%eval(&yr)%eval(&mn);  
%let k=%eval(&k+1);  
%end;  
%end;  
%let tot=%eval(&k-1);  
  
DATA OUT1.Y&SYSPARM;  
MERGE  
OLD.Y&SYSPARM  
FIN.LINKSSN (IN=INFIN)  
IN1.LONG(IN = INL KEEP=SSN MFT /* FOR SGA FOR BLIND */  
%DO I=1 %TO &TOT;  
EIN&&X&I FDA&&X&I  
FDP&&X&I STP&&X&I  
SPA&&X&I UIN&&X&I  
LVF&&X&I MDT&&X&I  
PST&&X&I STC&&X&I  
T16DX1&&X&I  
T16DX2&&X&I  
%END;
```

Appendix A.80
SAS Code: T2T16YR2

```
RENAME=(
%DO I=1 %TO &TOT;
    EIN&&X&I=EICM&&X&I FDA&&X&I=FAMT&&X&I
    FDP&&X&I=FPMT&&X&I STP&&X&I=SPMT&&X&I
    SPA&&X&I=SAMT&&X&I UIN&&X&I=UINC&&X&I
    LVF&&X&I=LIVF&&X&I MDT&&X&I=MTST&&X&I
    PST&&X&I=PSTA&&X&I STC&&X&I=SCON&&X&I
%END;))
IN2.MBR (IN=INM KEEP=SSN HI_START HI_TERM SMI_STAR SMI_TERM
%DO I=1 %TO &TOT; MBC&&X&I LAF&&X&I WIC&&X&I RFD&&X&I
RFST&&X&I BPD&&X&I DPEN&&X&I DUEO&&X&I PAYD&&X&I PAYO&&X&I
T2DX1&&X&I T2DX2&&X&I MBA&&X&I MBP&&X&I CDR&&X&I
DIRPAY&&X&I MEDPREM&&X&I
%END;
RENAME=(
%DO I=1 %TO &TOT; MBC&&X&I=DUED&&X&I
%END;))
IN3.ALLCDR
(IN=ALL KEEP=SSN
%DO I=1 %TO &TOT;
    ALX&&X&I
    EDX&&X&I
    MIEX&&X&I
%END;))
IN4.LDWSTRNG
( KEEP=SSN
%DO I=1 %TO &TOT;
    LDWDI&&X&I
    LDWSSI&&X&I
    LDWCM&&X&I
%END;))
IN5.T2EARN
( KEEP=COSSN
%DO I=1 %TO &TOT;
    T2GRSAMT&&X&I
    T2NETAMT&&X&I
    T2SEHRS&&X&I
    T2SEVERIND&&X&I
    T2VERIND&&X&I
%END;
RENAME=( COSSN=SSN ))
IN6.T2WKDETN
( KEEP=COSSN
%DO I=1 %TO &TOT;
    TWPDATA&&X&I
    T2CDNAMT&&X&I
    T2EXPAMT&&X&I
    T2FRAUDVER&&X&I
    T2SBDYAMT&&X&I
    T2UBEAMT&&X&I
%END;
RENAME=( COSSN=SSN ))
IN7.T16EARN
```

Appendix A.80
SAS Code: T2T16YR2

```

      ( KEEP=COSSN
%DO I=1 %TO &TOT;
  T16EXLAMT&&X&I
  T16EXPAMT&&X&I
  T16GRSAMT&&X&I
  T16NETAMT&&X&I
  T16PASAMT&&X&I
  T16SEVERIND&&X&I
  T16VERIND&&X&I
%END;
  RENAME=(COSSN=SSN))
IN8.ALGEARN
  ( KEEP=COSSN
%DO I=1 %TO &TOT;
  ALLGAMT&&X&I
%END;
  RENAME=(COSSN=SSN))
;
BY SSN;

/* DELETE CASES NOT IN THE MASTER FINDER */

IF INFIN ;

ARRAY FPMT(*) %DO I=1 %TO &TOT; FPMT&&X&I %END;;
ARRAY FAMT(*) %DO I=1 %TO &TOT; FAMT&&X&I %END;;
ARRAY PSTA(*) %DO I=1 %TO &TOT; PSTA&&X&I %END;;
ARRAY MTST(*) %DO I=1 %TO &TOT; MTST&&X&I %END;;
ARRAY SPMT(*) %DO I=1 %TO &TOT; SPMT&&X&I %END;;
ARRAY SAMT(*) %DO I=1 %TO &TOT; SAMT&&X&I %END;;
ARRAY SCON(*) %DO I=1 %TO &TOT; SCON&&X&I %END;;
ARRAY DBEN(*) %DO I=1 %TO &TOT; DUED&&X&I %END;;
ARRAY LAFCODE(*) %DO I=1 %TO &TOT; LAF&&X&I %END;;

ARRAY SBEN(*) %DO I=1 %TO &TOT; PAYS&&X&I %END;;
ARRAY SDUE(*) %DO I=1 %TO &TOT; DUES&&X&I %END;;

/* REVISION 2009 - TAKE OUT THE 0 */

DO I = 1 TO DIM(FPMT);
  SBEN(I) = SUM(OF FPMT(I),SPMT(I));
  SDUE(I) = SUM(OF FAMT(I),SAMT(I));
END;

/* CONCURRENT BENEFICIARY INDICATOR */
/* REVISION 2008 - INDICATOR WILL FLAG BENEFICIARIES RECEIVING
  A PAYMENT (IN CURRENT PAY) IN BOTH PROGRAMS, SSI AND SSDI */
ARRAY CONC(*) %DO I=1 %TO &TOT; CONC&&X&I %END;;

DO I = 1 TO DIM(DBEN);
  IF (SDUE(I) > 0 AND PSTA(I) IN ('C01' 'M01' 'M02')) AND
    (DBEN(I) > 0 AND SUBSTR(LAFCODE(I),1,1) IN ('C' 'E'))
    THEN CONC(I) = 1;

```

Appendix A.80
SAS Code: T2T16YR2

```
ELSE CONC(I) = 0;
END;

/* CREATE TERMINATION INDICATOR */
ARRAY TERMA (*) %DO I=1 %TO &TOT; TSSI&&X&I %END;;
ARRAY TERMB (*) %DO I=1 %TO &TOT; TSSD&&X&I %END;;

/* CREATE TERMINATION INDICATOR FOR SSI */

TERMSI=0;
DO I = 1 TO DIM(PSTA);
  IF SUBSTR(PSTA(I),1,1) = 'T' THEN TERMA(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN TERMA(I) = 0;
  IF TERMA(I)=1 THEN TERMSI=1; /* WILL BE WRITTEN FOR EACH YEAR */
END;

/* CREATE TEMINATION INDICATOR FOR SSDI */

TERMSSD=0;
DO I = 1 TO DIM(LAFCODE);

IF SUBSTR(LAFCODE(I),1,1) IN ('T','X') THEN TERMB(I)=1;
ELSE IF LAFCODE(I) > ' ' THEN TERMB(I)=0;
  IF TERMB(I)=1 THEN TERMSSD=1;
END;

/* CREATE MONTHLY INDICATORS FOR 1619A-1619B*/

ARRAY PROA(*) %DO I=1 %TO &TOT; PROA&&X&I %END;;
ARRAY PROB(*) %DO I=1 %TO &TOT; PROB&&X&I %END;;
ARRAY EICM(*) %DO I=1 %TO &TOT; EICM&&X&I %END;;

/* PROB */
/* REVISION 2009 - CREATE ONLY FOR SSI */

DO I = 1 TO DIM(MTST);
  IF MTST(I) IN ('A','B','F') AND
  PSTA(I) IN ('N01', 'P01', 'E01') AND
  EICM(I) > 0
  THEN PROB(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN PROB(I) = 0;
END;
DO I = 1 TO DIM(SCON);
/*FOR PROA (THIS SHOULD YIELD NO CASES WHERE PROB = 1 AND PROA =
1)*/
IF SCON(I) IN ('D','E','F','G') AND
PSTA(I) IN ('C01', 'M01', 'M02') AND
SDUE(I) > 0 AND
EICM(I) > 0
THEN PROA(I) = 1;
ELSE IF PSTA(I) NE ' ' THEN PROA(I) = 0;
END;
```

Appendix A.80

SAS Code: T2T16YR2

```

/* CREATE A MEDICARE INDICATOR */

ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;
ARRAY MEDR (*) %DO I=1 %TO &TOT; MEDR&&X&I %END;;
J=1;
Y=&BEGYR;
DO I = 1 TO DIM(CYEAR);
  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
  IF INT(I/12)=I/12 THEN DO;
    J=1;
    Y=Y+1;
  END;
  ELSE J=J+1;

  NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4) || SUBSTR(CYEAR(I),5,2),YYMMN6.);

  MEDR(I) = 0;
  IF (HI_START > .Z AND HI_START <= NYEAR(I)) OR
    (SMI_STAR > .Z AND SMI_STAR <= NYEAR(I) <= SMI_TERM)
    THEN MEDR(I) = 1;
  ELSE IF (HI_START > .Z AND (NYEAR(I) <= HI_START OR (HI_TERM > .Z
AND
  NYEAR(I) >= HI_TERM))) OR (SMI_STAR > .Z AND (NYEAR(I) <=
SMI_STAR
  OR (SMI_TERM > .Z AND NYEAR(I) >= SMI_TERM))) THEN MEDR(I) = 0;
  ELSE MEDR(I) = .;

END;

DROP Y J I %DO I=1 %TO &TOT; C&&X&I N&&X&I %END;;
DROP HI_START HI_TERM SMI_STAR SMI_TERM;

ARRAY LONE (*) %DO I=1 %TO &TOT; LONE&&X&I %END;;
ARRAY OTHR (*) %DO I=1 %TO &TOT; OTHR&&X&I %END;;
ARRAY FACL (*) %DO I=1 %TO &TOT; FACL&&X&I %END;;
ARRAY LIVF (*) %DO I=1 %TO &TOT; LIVF&&X&I %END;;

DO I = 1 TO DIM(LIVF);
  IF LIVF(I) = 'A' THEN LONE(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN LONE(I) = 0;
  ELSE LONE(I) = .;

  IF LIVF(I) = 'B' THEN OTHR(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN OTHR(I) = 0;
  ELSE OTHR(I) = .;

  IF LIVF(I) = 'D' THEN FACL(I) = 1;
  ELSE IF LIVF(I) > ' ' THEN FACL(I) = 0;
  ELSE FACL(I) = .;
END;

```

Appendix A.80
SAS Code: T2T16YR2

```
LENGTH
%DO I=1 %TO &TOT;
MBA&&X&I
MBP&&X&I
DUED&&X&I
PAYD&&X&I
PAYO&&X&I
DUEO&&X&I
FAMT&&X&I
FPMT&&X&I
SAMT&&X&I
PAYS&&X&I
DUES&&X&I
SPMT&&X&I
UINC&&X&I
TSSI&&X&I
TSSD&&X&I
CONC&&X&I
OTHR&&X&I
PROA&&X&I
LONE&&X&I
FACL&&X&I
MEDR&&X&I
DIRPAY&&X&I
MEDPREM&&X&I
%END;
4
TERMSSD TERMSSI 3
;
```

```
LABEL
TERMSSI = 'TERMINATION INDICATOR T16 FOR YEAR'
TERMSSD = 'TERMINATION INDICATOR T2 FOR YEAR'
%DO I=1 %TO &TOT;
DUEO&&X&I = "&&X&I DEPENDENT PAYMENT DUE"
DPEN&&X&I = "&&X&I NUMBER OF DEPENDENTS"
EICM&&X&I = "&&X&I COUNTABLE EARNED INCOME"
CONC&&X&I = "&&X&I CONCURRENTLY IN CURRENT PAY"
DUED&&X&I = "&&X&I FEDERAL SSDI BENEFIT CREDITED"
PAYD&&X&I = "&&X&I PHUS SSDI BENEFIT PAID"
PAYO&&X&I = "&&X&I PHUS SSDI DEPENDENT BENEFIT PAID"
MBA&&X&I = "&&X&I FEDERAL SSDI BENEFIT DUE"
MBP&&X&I = "&&X&I FEDERAL SSDI BENEFIT PAID"
FACL&&X&I = "&&X&I LIVES IN MEDICAL FACILITY"
FAMT&&X&I = "&&X&I FEDERAL SSI BENEFIT DUE"
FPMT&&X&I = "&&X&I FEDERAL SSI BENEFIT PAID"
LIVF&&X&I = "&&X&I LIVING ARRANGMENT"
LONE&&X&I = "&&X&I LIVES ALONE"
OTHR&&X&I = "&&X&I LIVES W/ ANOTHER SSI RECIPIENT"
PROA&&X&I = "&&X&I CONTINUING SSI ELIG - 1619(A)"
PSTA&&X&I = "&&X&I PAYMENT STATUS"
MTST&&X&I = "&&X&I INCOME TEST"
SAMT&&X&I = "&&X&I STATE SSI BENEFIT DUE"
```

Appendix A.80
SAS Code: T2T16YR2

```
PAYS&&X&I = "&&X&I SSI BENEFIT PAID"
SCON&&X&I = "&&X&I STATE CONCURRENT ELIGIBILITY IND"
DUES&&X&I = "&&X&I SSI BENEFIT DUE"
SPMT&&X&I = "&&X&I STATE SSI BENEFIT PAID"
UINC&&X&I = "&&X&I UNEARNED INCOME"
LAF&&X&I = "&&X&I LEDGER ACCOUNT FILE STATUS"
TSSI&&X&I = "&&X&I TERMINATED STATUS - T16"
TSSD&&X&I = "&&X&I TERMINATED STATUS - T2"
MEDR&&X&I = "&&X&I MEDICARE INDICATOR-T2"
DIRPAY&&X&I = "&&X&I PHUS DIRECT PAY"
MEDPREM&&X&I = "&&X&I PHUS MEDICARE PREMIUM"
%END;

SSN      = 'SOCIAL SECURITY NUMBER'
;

/* CREATE DIAGNOSIS VARIABLES */

ARRAY T2D1(*) %DO I = 1 %TO &TOT; T2DX1&&X&I %END;;
ARRAY T2D2(*) %DO I = 1 %TO &TOT; T2DX2&&X&I %END;;
ARRAY T16D1(*) %DO I = 1 %TO &TOT; T16DX1&&X&I %END;;
ARRAY T16D2(*) %DO I = 1 %TO &TOT; T16DX2&&X&I %END;;
ARRAY CDX1(*) $4 %DO I = 1 %TO &TOT; DX1X&&X&I %END;;
ARRAY CDX2(*) $4 %DO I = 1 %TO &TOT; DX2X&&X&I %END;;

/* COLLAPSE THE DIAGNOSIS CODES TO 1 CODE -
T2 TAKES PRECEDENCE IF BOTH ARE PRESENT */

/* BOTH T2 AND CONCS */
DO I = 1 TO DIM(T2D1);
  CDX1(I) = T2D1(I);
  CDX2(I) = T2D2(I);
END;

/* T16 ONLIES */
DO I = 1 TO DIM(T2D1);
  IF CDX1(I) = ' ' THEN CDX1(I) = T16D1(I);
  IF CDX2(I) = ' ' THEN CDX2(I) = T16D2(I);
END;

DROP %DO I = 1 %TO &TOT; T2DX1&&X&I T16DX1&&X&I
      T2DX2&&X&I T16DX2&&X&I %END;;
DROP I MFT;

LABEL
%DO I=1 %TO &TOT;
PROB&&X&I = "&&X&I CONTINUING MEDICAID ELIG - 1619(B)"
LDWSSI&&X&I = "&&X&I SSI LEFT DUE TO WORK INDICATOR"
LDWDI&&X&I = "&&X&I SSDI LEFT DUE TO WORK INDICATOR"
LDWCM&&X&I = "&&X&I COMBINED LEFT DUE TO WORK INDICATOR"
ZIP&&X&I = "&&X&I ZIP CODE (ZIPS)"
```

Appendix A.80
SAS Code: T2T16YR2

```
IUA1&&X&I = "&&X&I UNEARNED INCOME AMOUNT, SSDI"  
IUA3&&X&I = "&&X&I UNEARNED INCOME AMOUNT, WORKERS' COMPENSATION"  
IUA6&&X&I = "&&X&I UNEARNED INCOME AMOUNT, TANF"  
DX1X&&X&I = "&&X&I PRIMARY DISABLING CONDITION, MONTHLY"  
DX2X&&X&I = "&&X&I SECONDARY DISABLING CONDITION, MONTHLY"  
%END;  
;  
DROP I;  
RUN;  
%MEND;  
%START;  
  
proc contents DATA=OUT1.Y&SYSPARM;  
title "CONTENTS OF YEAR &SYSPARM";  
run;  
PROC FREQ DATA=OUT1.Y&SYSPARM; TABLES LAF: PSTA: LDWDI: LDWSSI:  
LDWCM: /LIST MISSING;  
RUN;  
PROC MEANS DATA=OUT1.Y&SYSPARM; VAR  
DUED:  
PAYD:  
PAYO:  
DUEO:  
FAMT:  
FPMT:  
SAMT:  
PAYS:  
DUES:  
SPMT:  
UINC:  
EICM:  
PROA:  
PROB:  
DIRPAY:  
MEDPREM:  
;  
RUN;  
  
PROC PRINT DATA=OUT1.Y&SYSPARM(FIRSTOBS=10000 OBS=10025); RUN;
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```

//#2127Y10 JOB (12510000,T715,,SAS,,ITC9FL),2127PAGE,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#2127
//*****
//*
//*      *-----*
//*      |   OPDR.TG.PRD.ETTW.#2127.TRF10.ANN.PRDLIB(T2T16Y10) |
//*      *-----*
//*
//* CREATE YEARLY FILES FOR 2010 WITH MONTHLY FIELDS FROM JAN TO DEC
//* DCF VARIABLES NOT INCLUDED BEFORE 2000
//* CREATED BY: MIRIAM LOEWENBERG 7/13/06
//* MODIFIED BY: JEREMY PAGE 2/8/12 FOR TRF10
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*****
//JS030    EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2010"'
//*
//IN1     DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2     DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN3     DD DSN=OPDR.TG.PRD.ETTW.N8043.TRF10P.ALLCDR2.SA.V1,DISP=SHR
//IN4     DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10.LDWSTRNG.SA.V1,DISP=SHR
//IN5     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2EARN.SA.V1,DISP=SHR
//IN6     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDET.N.SA.V1,DISP=SHR
//IN7     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//IN8     DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.ALGEARN.SA.V1,DISP=SHR
//IN9     DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBSORD.D1012.SSD1,DISP=SHR
//IN10    DD DSN=OPDR.TG.PRD.ETTW.N8043.COMBDBAD.D1012.SSD,DISP=SHR
//FIN     DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//TEMP    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          UNIT=TSILO
//OUT1    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2010.SA.V1,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN   DD *
OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY OBS=MAX;
*** CHANGE ENDYR AND ENDMN FOR EACH RUN;
%LET YR = SUBSTR(&SYSPARM,3,2);
%let begyr=&SYSPARM;
%let endyr=&SYSPARM; /* change as needed */
%let endmn=12;      /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);

```

Appendix A.81
JCL/SAS Code: T2T16Y10

```
%if &i=&endyr %then %let emn=%eval(&endmn);
%else %let emn=12;
%do j=1 %to &emn;
  %if &j<10 %then %let mn=0%eval(&j);
  %else %let mn=%eval(&j);
  %let x&k=%eval(&yr)%eval(&mn);
  %let k=%eval(&k+1);
%end;
%end;
%let tot=%eval(&k-1);

/* DEDUP THE DBAD FILE BEFORE MERGING */
PROC SORT DATA=IN10.DBAD NODUPKEY OUT=TEMP.DBAD
          (KEEP=SSN %DO I=1 %TO &TOT;
           ZIP&&X&I
           %END;);
          BY SSN;
RUN;

DATA OUT1.Y&SYSPARM;
MERGE
  FIN.LINKSSN (IN=INFIN)
  IN1.LONG (IN=INL KEEP=SSN MFT /* FOR SGA FOR BLIND */
           %DO I=1 %TO &TOT;
             EIN&&X&I FDA&&X&I
             FDP&&X&I STP&&X&I
             SPA&&X&I UIN&&X&I
             LVF&&X&I MDT&&X&I
             PST&&X&I STC&&X&I
             T16DX1&&X&I
             T16DX2&&X&I
           %END;
           RENAME=(
             %DO I=1 %TO &TOT;
               EIN&&X&I=EICM&&X&I FDA&&X&I=FAMT&&X&I
               FDP&&X&I=FPMT&&X&I STP&&X&I=SPMT&&X&I
               SPA&&X&I=SAMT&&X&I UIN&&X&I=UINC&&X&I
               LVF&&X&I=LIVF&&X&I MDT&&X&I=MTST&&X&I
               PST&&X&I=PSTA&&X&I STC&&X&I=SCON&&X&I
             %END;))
  IN2.MBR (IN=INM KEEP=SSN HI_START HI_TERM SMI_STAR SMI_TERM
           %DO I=1 %TO &TOT;
             MBC&&X&I LAF&&X&I WIC&&X&I RFD&&X&I
             RFST&&X&I BPD&&X&I DPEN&&X&I DUEO&&X&I PAYD&&X&I
             PAYO&&X&I T2DX1&&X&I T2DX2&&X&I MBA&&X&I MBP&&X&I
             CDR&&X&I MEDPREM&&X&I DIRPAY&&X&I
           %END;
           RENAME=(
             %DO I=1 %TO &TOT; MBC&&X&I=DUED&&X&I
             %END;))
  IN3.ALLCDR
          (IN=ALL KEEP=SSN
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```
%DO I=1 %TO &TOT;
    ALX&&X&I
    EDX&&X&I
    MIE&&X&I
%END; )
IN4.LDWSTRNG
    ( KEEP=SSN
%DO I=1 %TO &TOT;
    LDWDI&&X&I
    LDWSSI&&X&I
    LDWCM&&X&I
%END; )
IN5.T2EARN
    ( KEEP=COSSN
%DO I=1 %TO &TOT;
    T2GRSAMT&&X&I
    T2NETAMT&&X&I
    T2SEHRS&&X&I
    T2SEVERIND&&X&I
    T2VERIND&&X&I
%END;
    RENAME=( COSSN=SSN ) )
IN6.T2WKDETN
    ( KEEP=COSSN
%DO I=1 %TO &TOT;
    TWPDATA&&X&I
    T2CDNAMT&&X&I
    T2EXPAMT&&X&I
    T2FRAUDVER&&X&I
    T2SBDYAMT&&X&I
    T2UBEAMT&&X&I
%END;
    RENAME=( COSSN=SSN ) )
IN7.T16EARN
    ( KEEP=COSSN
%DO I=1 %TO &TOT;
    T16EXLAMT&&X&I
    T16EXPAMT&&X&I
    T16GRSAMT&&X&I
    T16NETAMT&&X&I
    T16PASAMT&&X&I
    T16SEVERIND&&X&I
    T16VERIND&&X&I
%END;
    RENAME=( COSSN=SSN ) )
IN8.ALGEARN
    ( KEEP=COSSN
%DO I=1 %TO &TOT;
    ALLGAMT&&X&I
%END;
    RENAME=( COSSN=SSN ) )
IN9.SORD(IN=INR KEEP=PAN
%DO I=1 %TO &TOT;
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```
      IET1&&X&I IEA1&&X&I
      IET2&&X&I IEA2&&X&I
      IET3&&X&I IEA3&&X&I
      IET4&&X&I IEA4&&X&I
      IET5&&X&I IEA5&&X&I
      IET6&&X&I IEA6&&X&I
      MEDC&&X&I IUA1&&X&I IUA3&&X&I IUA6&&X&I
      ST&&X&I
      CNTY&&X&I
      %END;
      RENAME=(PAN=SSN)
TEMP.DBAD (IN=INZIP KEEP=SSN
          %DO I=1 %TO &TOT;
            ZIP&&X&I
          %END;)
;
      BY SSN;
/* DELETE CASES NOT IN THE MASTER FINDER */
      IF INFIN;

ARRAY FPMT(*) %DO I=1 %TO &TOT; FPMT&&X&I %END;;
ARRAY FAMT(*) %DO I=1 %TO &TOT; FAMT&&X&I %END;;
ARRAY PSTA(*) %DO I=1 %TO &TOT; PSTA&&X&I %END;;
ARRAY MTST(*) %DO I=1 %TO &TOT; MTST&&X&I %END;;
ARRAY SPMT(*) %DO I=1 %TO &TOT; SPMT&&X&I %END;;
ARRAY SAMT(*) %DO I=1 %TO &TOT; SAMT&&X&I %END;;
ARRAY SCON(*) %DO I=1 %TO &TOT; SCON&&X&I %END;;
ARRAY DBEN(*) %DO I=1 %TO &TOT; DUED&&X&I %END;;
ARRAY LAFCODE(*) %DO I=1 %TO &TOT; LAF&&X&I %END;;

ARRAY SBEN(*) %DO I=1 %TO &TOT; PAYS&&X&I %END;;
ARRAY SDUE(*) %DO I=1 %TO &TOT; DUES&&X&I %END;;

/* REVISION 2009- TAKE THE ZERO OUT OF THE CODE */

DO I = 1 TO DIM(FPMT);
  SBEN(I) = SUM(OF FPMT(I),SPMT(I));
  SDUE(I) = SUM(OF FAMT(I),SAMT(I));
END;

/* CONCURRENT BENEFICIARY INDICATOR */
/* REVISION 2008 - INDICATOR WILL FLAG BENEFICIARIES RECEIVING
  A PAYMENT (IN CURRENT PAY) IN BOTH PROGRAMS, SSI AND SSDI */
ARRAY CONC(*) %DO I=1 %TO &TOT; CONC&&X&I %END;;

DO I = 1 TO DIM(DBEN);
  IF (SDUE(I) > 0 AND PSTA(I) IN ('C01' 'M01' 'M02')) AND
     (DBEN(I) > 0 AND SUBSTR(LAFCODE(I),1,1) IN ('C' 'E'))
  THEN CONC(I) = 1;
  ELSE CONC(I) = 0;
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```
END;

/* CREATE TERMINATION INDICATOR */
ARRAY TERMA (*) %DO I=1 %TO &TOT; TSSI&&X&I %END;;
ARRAY TERMB (*) %DO I=1 %TO &TOT; TSSD&&X&I %END;;

/* CREATE TERMINATION INDICATOR FOR SSI */
TERMSSI=0;
DO I = 1 TO DIM(PSTA);
  IF SUBSTR(PSTA(I),1,1) = 'T' THEN TERMA(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN TERMA(I) = 0;
  IF TERMA(I)=1 THEN TERMSSI=1; /* WILL BE WRITTEN FOR EACH YEAR */
END;

/* CREATE TEMINATION INDICATOR FOR SSDI */
TERMSSD=0;
DO I = 1 TO DIM(LAFCODE);
  IF SUBSTR(LAFCODE(I),1,1) IN ('T','X') THEN TERMB(I)=1;
  ELSE IF LAFCODE(I) > ' ' THEN TERMB(I)=0;
  IF TERMB(I)=1 THEN TERMSSD=1;
END;

/* CREATE MONTHLY INDICATORS FOR 1619A-1619B*/
ARRAY PROA(*) %DO I=1 %TO &TOT; PROA&&X&I %END;;
ARRAY PROB(*) %DO I=1 %TO &TOT; PROB&&X&I %END;;
ARRAY EICM(*) %DO I=1 %TO &TOT; EICM&&X&I %END;;

/* PROB */
/* REVISION 2008 - CREATE ONLY FOR SSI */
DO I = 1 TO DIM(MTST);
  IF MTST(I) IN ('A','B','F') AND
    PSTA(I) IN ('N01', 'P01', 'E01') AND
    EICM(I) > 0
    THEN PROB(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN PROB(I) = 0;
END;

/*FOR PROA (THIS SHOULD YIELD NO CASES WHERE PROB = 1 AND PROA =
1)*/
DO I = 1 TO DIM(SCON);
  IF SCON(I) IN ('D','E','F','G') AND
    PSTA(I) IN ('C01', 'M01', 'M02') AND
    SDUE(I) > 0 AND
    EICM(I) > 0
    THEN PROA(I) = 1;
  ELSE IF PSTA(I) NE ' ' THEN PROA(I) = 0;
END;

/* CREATE A MEDICARE INDICATOR */
ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;
ARRAY MEDR (*) %DO I=1 %TO &TOT; MEDR&&X&I %END;;
J=1;
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```

Y=&BEGYR;
DO I = 1 TO DIM(CYEAR);
  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
  IF INT(I/12)=I/12 THEN DO;
    J=1;
    Y=Y+1;
  END;
  ELSE J=J+1;

  NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4) || SUBSTR(CYEAR(I),5,2),YYMMN6.);

  MEDR(I) = 0;
  IF (HI_START > .Z AND HI_START <= NYEAR(I)) OR
    (SMI_STAR > .Z AND SMI_STAR <= NYEAR(I) <= SMI_TERM)
    THEN MEDR(I) = 1;
  ELSE IF (HI_START > .Z AND (NYEAR(I) <= HI_START OR (HI_TERM > .Z
AND
  NYEAR(I) >= HI_TERM))) OR (SMI_STAR > .Z AND (NYEAR(I) <=
SMI_STAR
  OR (SMI_TERM > .Z AND NYEAR(I) >= SMI_TERM))) THEN MEDR(I) = 0;
  ELSE MEDR(I) = .;

END;

DROP Y J I %DO I=1 %TO &TOT; C&&X&I N&&X&I %END;;
DROP HI_START HI_TERM SMI_STAR SMI_TERM;

ARRAY LONE (*) %DO I=1 %TO &TOT; LONE&&X&I %END;;
ARRAY OTHR (*) %DO I=1 %TO &TOT; OTHR&&X&I %END;;
ARRAY FACL (*) %DO I=1 %TO &TOT; FACL&&X&I %END;;
ARRAY LIVF (*) %DO I=1 %TO &TOT; LIVF&&X&I %END;;

DO I = 1 TO DIM(LIVF);
  IF LIVF(I) = 'A' THEN LONE(I) = 1;
  ELSE IF LIVF(I) NE ' ' THEN LONE(I) = 0;
  ELSE LONE(I) = .;

  IF LIVF(I) = 'B' THEN OTHR(I) = 1;
  ELSE IF LIVF(I) NE ' ' THEN OTHR(I) = 0;
  ELSE OTHR(I) = .;

  IF LIVF(I) = 'D' THEN FACL(I) = 1;
  ELSE IF LIVF(I) NE ' ' THEN FACL(I) = 0;
  ELSE FACL(I) = .;
END;

LENGTH
%DO I=1 %TO &TOT;
MBA&&X&I
MBP&&X&I
DUED&&X&I
PAYD&&X&I

```

Appendix A.81
JCL/SAS Code: T2T16Y10

PAYO&&X&I
DUEO&&X&I
FAMT&&X&I
FPMT&&X&I
SAMT&&X&I
PAYS&&X&I
DUES&&X&I
SPMT&&X&I
UINC&&X&I
TSSI&&X&I
TSSD&&X&I
CONC&&X&I
OTHR&&X&I
PROA&&X&I
LONE&&X&I
FACL&&X&I
MEDR&&X&I
DIRPAY&&X&I
MEDPREM&&X&I

%END;

4

TERMSSD TERMSSI 3

;

LABEL

TERMSSI = 'TERMINATION INDICATOR T16 FOR YEAR'
TERMSSD = 'TERMINATION INDICATOR T2 FOR YEAR'
%DO I=1 %TO &TOT;
DUEO&&X&I = "&&X&I DEPENDENT PAYMENT DUE"
DPEN&&X&I = "&&X&I NUMBER OF DEPENDENTS"
EICM&&X&I = "&&X&I COUNTABLE EARNED INCOME"
CONC&&X&I = "&&X&I CONCURRENTLY IN CURRENT PAY"
DUED&&X&I = "&&X&I FEDERAL SSDI BENEFIT CREDITED"
PAYD&&X&I = "&&X&I PHUS SSDI BENEFIT PAID"
PAYO&&X&I = "&&X&I PHUS SSDI DEPENDENT BENEFIT PAID"
MBA&&X&I = "&&X&I FEDERAL SSDI BENEFIT DUE"
MBP&&X&I = "&&X&I FEDERAL SSDI BENEFIT PAID"
FACL&&X&I = "&&X&I LIVES IN MEDICAL FACILITY"
FAMT&&X&I = "&&X&I FEDERAL SSI BENEFIT DUE"
FPMT&&X&I = "&&X&I FEDERAL SSI BENEFIT PAID"
LIVF&&X&I = "&&X&I LIVING ARRANGMENT"
LONE&&X&I = "&&X&I LIVES ALONE"
OTHR&&X&I = "&&X&I LIVES W/ ANOTHER SSI RECIPIENT"
PROA&&X&I = "&&X&I CONTINUING SSI ELIG - 1619(A)"
PSTA&&X&I = "&&X&I PAYMENT STATUS"
MTST&&X&I = "&&X&I INCOME TEST"
SAMT&&X&I = "&&X&I STATE SSI BENEFIT DUE"
PAYS&&X&I = "&&X&I SSI BENEFIT PAID"
SCON&&X&I = "&&X&I STATE CONCURRENT ELIGIBILITY IND"
DUES&&X&I = "&&X&I SSI BENEFIT DUE"
SPMT&&X&I = "&&X&I STATE SSI BENEFIT PAID"
UINC&&X&I = "&&X&I UNEARNED INCOME"
LAF&&X&I = "&&X&I LEDGER ACCOUNT FILE STATUS"

Appendix A.81
JCL/SAS Code: T2T16Y10

```
TSSI&&X&I = "&&X&I TERMINATED STATUS - T16"  
TSSD&&X&I = "&&X&I TERMINATED STATUS - T2"  
MEDR&&X&I = "&&X&I MEDICARE INDICATOR-T2"  
DIRPAY&&X&I = "&&X&I PHUS DIRECT PAY"  
MEDPREM&&X&I = "&&X&I PHUS MEDICARE PREMIUM"  
%END;
```

```
SSN      = 'SOCIAL SECURITY NUMBER'  
;
```

```
/* CREATE MONTHLY STATE INDICATORS FOR PHASE CONSTRUCTION */  
LENGTH  
%DO I=1 %TO &TOT;  
    PST&&X&I  
%END; $2;
```

```
ARRAY ZP(*) $ %DO I=1 %TO &TOT; ZIP&&X&I %END;;  
ARRAY RZP(*) $ %DO I=1 %TO &TOT; ST&&X&I %END;;  
ARRAY ST(*) $ %DO I=1 %TO &TOT; PST&&X&I %END;;
```

```
DO I = 1 TO DIM(RZP);  
    IF RZP(I) NE ' ' THEN ST(I) = FIPSTATE(RZP(I));  
END;  
/* IF NO VALUE FOR SSI USE DI */  
DO I = 1 TO DIM(ZP);  
    IF ST(I) = ' ' AND ZP(I) NE ' ' THEN ST(I) = ZIPSTATE(ZP(I));  
END;
```

```
/* CREATE MONTHLY INDICATORS FOR IRWE PASS AND SELF-EMPLOY - REMICS  
*/
```

```
ARRAY IE5T(*) $ %DO I=1 %TO &TOT; IET5&&X&I %END;;  
ARRAY IE2T(*) $ %DO I=1 %TO &TOT; IET2&&X&I %END;;  
ARRAY IE4T(*) $ %DO I=1 %TO &TOT; IET4&&X&I %END;;  
ARRAY IIND(*) %DO I=1 %TO &TOT; IIND&&X&I %END;;  
ARRAY PIND(*) %DO I=1 %TO &TOT; PIND&&X&I %END;;  
ARRAY SIND(*) %DO I=1 %TO &TOT; SIND&&X&I %END;;
```

```
DO I = 1 TO DIM(IIND);  
    IF IE5T(I) = 'T' THEN IIND(I) = 1;  
    ELSE IIND(I) = 0;  
  
    IF IE2T(I) = 'D' THEN PIND(I) = 1;  
    ELSE PIND(I) = 0;  
  
    IF IE4T(I) = 'S' THEN SIND(I) = 1;  
    ELSE SIND(I) = 0;  
END;
```

```
/* TRIAL WORK PERIOD INDICATOR MOVED TO DCF */  
ARRAY ERN1 (*) %DO I=1 %TO &TOT; EARN&&X&I %END;;  
ARRAY SELF (*) %DO I=1 %TO &TOT; IEA4&&X&I %END;;  
ARRAY WAGE (*) %DO I=1 %TO &TOT; IEA6&&X&I %END;;  
ARRAY EIND (*) %DO I=1 %TO &TOT; EIND&&X&I %END;;
```

Appendix A.81
JCL/SAS Code: T2T16Y10

```

ARRAY SGA  (*) %DO I=1 %TO &TOT; SGA&&X&I %END;;
ARRAY IEA4 (*) %DO I=1 %TO &TOT; IEA4&&X&I %END;;
ARRAY IEA6 (*) %DO I=1 %TO &TOT; IEA6&&X&I %END;;

DO I = 1 TO DIM(ERN1);
  ERN1(I) = .;
  IF SELF(I) > .Z OR WAGE(I) > .Z THEN DO;
    ERN1(I) = SUM(OF IEA4(I),IEA6(I));
  END;

  IF ERN1(I) > 0 THEN EIND(I) = 1;
  ELSE IF ERN1(I) = 0 THEN EIND(I) = 0;
  ELSE EIND(I) = .;

  /* SGA IS DIFFERENT FOR BLIND PEOPLE */
  /* REVISED CODE UPDATED EVERY YEAR - THIS IS COMPUTED ONLY FOR
  THE NEW YEAR OF DATA COLLECTION. UPDATED EARNINGS CEILINGS
  ARE INCLUDED. EARLIER YEARS ALREADY HAVE THE VARIABLE AND IT
  IS MERGED ON FROM THE OLD FILE */
  IF SUBSTR(MFT,1,1) = 'B' THEN DO; /* BLIND PEOPLE */
    IF ERN1(I) >= 1640 THEN SGA(I) = 1;
    ELSE IF 0 <= ERN1(I) < 1640 THEN SGA(I) = 0;
    ELSE SGA(I) = .;
  END;
  ELSE DO;
    IF ERN1(I) >= 1000 THEN SGA(I) = 1;
    ELSE IF 0 <= ERN1(I) < 1000 THEN SGA(I) = 0;
    ELSE SGA(I) = .;
  END;
END; /* LOOP THROUGH EARNINGS */

/*CREATE AN INDICATOR FOR 1619(B) */
ARRAY MEDE (*) %DO I=1 %TO &TOT; MEDE&&X&I %END;;
ARRAY MEDC (*) %DO I=1 %TO &TOT; MEDC&&X&I %END;;

DO I = 1 TO DIM(MEDC);
  IF MEDC(I) IN ('C','W','Y') THEN MEDE(I) = 1;
  ELSE IF MEDC(I) > ' ' THEN MEDE(I) = 0;
  ELSE MEDE(I) = .;
END;

/* CREATE DIAGNOSIS VARIABLES */
ARRAY T2D1(*) %DO I = 1 %TO &TOT; T2DX1&&X&I %END;;
ARRAY T2D2(*) %DO I = 1 %TO &TOT; T2DX2&&X&I %END;;
ARRAY T16D1(*) %DO I = 1 %TO &TOT; T16DX1&&X&I %END;;
ARRAY T16D2(*) %DO I = 1 %TO &TOT; T16DX2&&X&I %END;;
ARRAY CDX1(*) $4 %DO I = 1 %TO &TOT; DX1X&&X&I %END;;
ARRAY CDX2(*) $4 %DO I = 1 %TO &TOT; DX2X&&X&I %END;;

/* COLLAPSE THE DIAGNOSIS CODES TO 1 CODE -
T2 TAKES PRECEDENCE IF BOTH ARE PRESENT */

```

Appendix A.81
JCL/SAS Code: T2T16Y10

```

/* BOTH T2 AND CONCS */
DO I = 1 TO DIM(T2D1);
  CDX1(I) = T2D1(I);
  CDX2(I) = T2D2(I);
END;

/* T16 ONLY */
DO I = 1 TO DIM(T2D1);
  IF CDX1(I) = ' ' THEN CDX1(I) = T16D1(I);
  IF CDX2(I) = ' ' THEN CDX2(I) = T16D2(I);
END;

DROP %DO I = 1 %TO &TOT; T2DX1&&X&I T16DX1&&X&I
      T2DX2&&X&I T16DX2&&X&I %END;;
DROP I MFT;

LENGTH
%DO I=1 %TO &TOT;
IEA4&&X&I
IEA6&&X&I
EARN&&X&I
EIND&&X&I
SGA&&X&I
IIND&&X&I
PIND&&X&I
SIND&&X&I
PROB&&X&I
MEDE&&X&I
%END;
  4
;

LABEL
%DO I=1 %TO &TOT;
EARN&&X&I = "&&X&I EARNED INCOME (REMICS)"
EIND&&X&I = "&&X&I EMPLOYMENT INDICATOR"
IEA4&&X&I = "&&X&I INCOME FROM SELF-EMPLOYMENT"
IEA6&&X&I = "&&X&I INCOME FROM WAGES"
IET2&&X&I = "&&X&I EARNED INCOME TYPE (PASS)"
IET4&&X&I = "&&X&I EARNED INCOME TYPE (SELF EMPLOY)"
IET5&&X&I = "&&X&I EARNED INCOME TYPE (IRWE)"
IEA1&&X&I = "&&X&I EARNED INCOME BLIND"
IEA2&&X&I = "&&X&I EARNED INCOME PASS"
IEA3&&X&I = "&&X&I EARNED INCOME NET LOSS"
IEA5&&X&I = "&&X&I INCOME IRWE"
IET1&&X&I = "&&X&I EARNED INCOME TYPE (BLIND)"
IET3&&X&I = "&&X&I EARNED INCOME TYPE (NET LOSS)"
IET6&&X&I = "&&X&I EARNED INCOME TYPE (WAGES)"
IIND&&X&I = "&&X&I USE OF IRWE INDICATOR"
PIND&&X&I = "&&X&I USE OF PASS INDICATOR"
SGA&&X&I = "&&X&I SGA FLAG"
SIND&&X&I = "&&X&I SELF-EMPLOYMENT INDICATOR"
CNTY&&X&I = "&&X&I REMICS COUNTY OF RESIDENCE"

```

Appendix A.81
JCL/SAS Code: T2T16Y10

```
ST&&X&I      = "&&X&I REMICS STATE OF RESIDENCE"  
PST&&X&I      = "&&X&I STATE POSTAL CODE"  
MEDC&&X&I     = "&&X&I MEDICAID ENROLLMENT STATUS"  
MEDE&&X&I     = "&&X&I MEDICAID ELIGIBILITY"  
PROB&&X&I     = "&&X&I CONTINUING MEDICAID ELIG - 1619(B)"  
LDWSSI&&X&I   = "&&X&I SSI LEFT DUE TO WORK INDICATOR"  
LDWDI&&X&I    = "&&X&I SSDI LEFT DUE TO WORK INDICATOR"  
LDWCM&&X&I    = "&&X&I COMBINED LEFT DUE TO WORK INDICATOR"  
ZIP&&X&I      = "&&X&I ZIP CODE (ZIPS)"  
IUA1&&X&I     = "&&X&I UNEARNED INCOME AMOUNT, SSDI"  
IUA3&&X&I     = "&&X&I UNEARNED INCOME AMOUNT, WORKERS' COMPENSATION"  
IUA6&&X&I     = "&&X&I UNEARNED INCOME AMOUNT, TANF"  
DX1X&&X&I    = "&&X&I PRIMARY DISABLING CONDITION, MONTHLY"  
DX2X&&X&I    = "&&X&I SECONDARY DISABLING CONDITION, MONTHLY"  
%END;  
  ;  
DROP I;  
RUN;  
%MEND;  
%START;  
  
PROC CONTENTS DATA=OUT1.Y&SYSPARM;  
TITLE "CONTENTS OF YEAR Y&SYSPARM";  
RUN;  
PROC FREQ DATA=OUT1.Y&SYSPARM; TABLES LAF: PSTA: LDWDI: LDWSSI:  
  LDWCM: /LIST MISSING;  
RUN;  
PROC MEANS DATA=OUT1.Y&SYSPARM; VAR  
  DUED:  
  PAYD:  
  PAYO:  
  DUEO:  
  FAMT:  
  FPMT:  
  SAMT:  
  PAYS:  
  DUES:  
  SPMT:  
  UINC:  
  EICM:  
  PROA:  
  PROB:  
  DIRPAY:  
  MEDPREM:  
  ;  
RUN;  
  
PROC PRINT DATA=OUT1.Y&SYSPARM(FIRSTOBS=10000 OBS=10025);  
TITLE 'PRINT OF YEAR Y&SYSPARM'; RUN;
```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```

/* -----*/
/*      NAME:          PAYMENT.SAS                               */
/*      DATE CREATED:  MAY 28 2011                             */
/*      TASK           :  PROCESS EN PAYMENT DATA 6FOR MINITRF  */
/*      MODIFIED       :  March 16, 2012 Xiao Fu FOR TRF10      */
/* -----*/

LIBNAME INTER 'M:\Page\TRF10\ENPayments\Data\IntermediateData\PII';
LIBNAME OUT   'M:\TRF10\EN Payments\Data\PII';

OPTIONS NOCENTER LS=96 MACROGEN MPRINT COMPRESS=BINARY;
OPTIONS OBS=MAX;

%LET BEGYR=2002;
%LET ENDYR=2010;
%LET ENDMN=12;

%MACRO START;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

DATA PAYMENT O_PAYMENT M_PAYMENT;
  SET INTER.ENPAY (WHERE=(SSN NE ' ') DROP=BENE_NAME);

SSN = LEFT(TRIM(COMPRESS(SSN,'-')));
DUNS = LEFT(TRIM(COMPRESS(DUNS,'-')));
EN   = LEFT(TRIM(EN));

/* CREATE DATE VARIABLE FROM MONTH_S - SET DAY TO FIRST DAY OF THE
MONTH*/
/* TRANSPOSE THE MONTH_S FIELD WHICH CAME IN AS A WRONGLY FORMATTED
DATE */

/*~~~~~
| QA CHECK:
| CHECK THAT THE CODE BELOW IS NEEDED FOR EACH NEW ROUND OF TRF DATA.
| IN TRF10 THE PAYMENTS FILE WAS DELIVERED WITH THE VARIABLE 'MONTH-
| YR',RENAMED AS 'MONTH_S, FORMATED AS '10-DEC' TO INDICATE DECEMBER,
| 2010.WHEN THE DATA IS READ INTO SAS IN THE PREVIOUS PROGRAM, THE

```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```
| DATE IS CONVERTED INTO '10DEC2012'. TO GET THE CORRECT MONTH AND
| YEAR FROM THE VARIABLE WE EXTRACT THE DAY AND NAME IT 'YEAR' AND
| EXTRACT MONTH NORMALLY
~~~~~*;/
```

```
YEAR = DAY(MONTH_S);
MONTH = MONTH(MONTH_S);
MONTHDATE = MDY(MONTH,1,YEAR);
```

```
/* THERE ARE SOME "PHI" CODES IN THE PAYMENT_NUMBER FIELD.
THESE HAVE BEEN RECODED IN THE PAYMENT_NUMBER_SORT FIELD BECAUSE
THEY ARE REALLY 'M' CODES. THEREFORE WE WILL USE THE RECODED
FIELD.PHI CODES ARE AUTOMATED PAYMENTS. THERE ARE SOME B CODES
THAT ARE BLIND M CODES. WHEN PAYMENT TYPE IS MO THEN LOOK AT
PAYMENT NUMBER FIELD TO SELECT TYPE */
```

```
LENGTH PAYMENT_TYPE $1;
TYPE_PAYMENT_NUMBER = SUBSTR(Pay_Num,1,1);
```

```
IF PYMT_TYPE = 'MO' THEN DO;
  IF TYPE_PAYMENT_NUMBER = 'P' THEN PAYMENT_TYPE = 'M';
  ELSE IF TYPE_PAYMENT_NUMBER = 'O' THEN PAYMENT_TYPE = 'O';
END;
```

```
ELSE IF TYPE_PAYMENT_NUMBER = 'O' THEN PAYMENT_TYPE = 'O';
```

```
/* A FEW RECORDS HAVE PAYMENT DATE EARLIER THAN THE TRIGGERING
MONTH. RECODE THE PAYMENT DATE TO THE MONTH DATE */
IF DATE_PAYMENT LT MONTHDATE THEN DATE_PAYMENT = MONTHDATE;
```

```
IF PAYMENT_TYPE = 'O' THEN OUTPUT O_PAYMENT;
ELSE IF PAYMENT_TYPE = 'M' THEN OUTPUT M_PAYMENT;
```

```
OUTPUT PAYMENT;
RUN;
```

```
PROC FREQ DATA=PAYMENT; TABLES
PYMT_TYPE*TYPE_PAYMENT_NUMBER*PAYMENT_TYPE/LIST MISSING; RUN;
```

```
/* SET ASIDE THE LIST OF EN'S FOR BENEFICIARIES TO MERGE ON LATER -
THERE IS MORE THAN 1 EN PER BENEFICIARY
WHERE SPLIT PAYMENTS ARE INDICATED THERE WILL BE GT 1 EN'S */
```

```
PROC SORT DATA=PAYMENT (KEEP=SSN EN DUNS) OUT=ENDATA NODUPKEY ;
BY SSN DUNS; RUN;
```

```
/* MAKE A MACRO VARIABLE FOR THE MAXIMUM NUMBER OF EN'S */
DATA CHECK (KEEP=COUNTENS);
  SET ENDATA; BY SSN DUNS;
```

```
RETAIN COUNTENS;
IF FIRST.SSN THEN COUNTENS = 1;
ELSE COUNTENS + 1;
```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```
IF LAST.SSN THEN OUTPUT;
RUN;

PROC FREQ; TABLES COUNTENS; RUN;

PROC MEANS DATA=CHECK NOPRINT;
VAR COUNTENS;
OUTPUT OUT=LAST (DROP=_)
      MAX=MAXENS;
RUN;

PROC PRINT; RUN;

DATA _NULL_;
SET LAST;
CALL SYMPUT('INX',MAXENS);
STOP;
RUN;

/*PROCESS PAYMENTS DATA */
DATA ENDATA (KEEP=SSN EN_NAME: DUNS_ID:);
SET ENDATA; BY SSN DUNS;

RETAIN
  %DO I = 1 %TO &INX;
    EN_NAME&I
    DUNS_ID&I
  %END;;

LENGTH
  %DO I = 1 %TO &INX;
    EN_NAME&I
  %END;
  $80;

LENGTH
  %DO I = 1 %TO &INX;
    DUNS_ID&I
  %END;
  $13;

ARRAY NAME (*) $ %DO I = 1 %TO &INX; EN_NAME&I %END;;
ARRAY NUMB (*) $ %DO I = 1 %TO &INX; DUNS_ID&I %END;;

IF FIRST.SSN THEN DO;
  %DO I = 1 %TO &INX;
    EN_NAME&I = ' ';
    DUNS_ID&I = ' ';
  %END;;
REC = 1;
NAME(REC) = EN;
***ADDING LEADING ZEROS TO DUNS_ID;
NUMB(REC) = PUT(DUNS,Z9.);
```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```
END;

ELSE DO;
  REC + 1; /*IMPLICIT RETAIN */
  NAME(REC) = EN;
  ***ADDING LEADING ZEROS TO DUNS_ID;
  NUMB(REC) = PUT(DUNS,Z9.);
END;

IF LAST.SSN THEN OUTPUT;
RUN;

/*SORT DATA BY SSN AND MONTHDATE */
PROC SORT DATA=O_PAYMENT; BY SSN MONTHDATE; RUN;
PROC SORT DATA=M_PAYMENT; BY SSN MONTHDATE; RUN;

/* THE FOLLOWING MACRO FIRST SUMS THE PAYMENT AMOUNTS FOR
PAYMENTS TRIGGERED ON THE SAME DATE. THEN IT CREATES
MONTHLY VARIABLES WHERE APPLICABLE AND OUTPUTS 1 RECORD
FOR EACH BENEFICIARY */

%MACRO SUMPAY (TYPE);

/* RESOLVE ALL DUPLICATE RECORDS */
DATA &TYPE._PAYMENT;
  SET &TYPE._PAYMENT; BY SSN MONTHDATE;

RETAIN AMTPAID;
IF FIRST.MONTHDATE AND LAST.MONTHDATE THEN DO;
  AMTPAID = AMOUNT;
END;

  /* PROCESS REMAINING DUPLICATES */
ELSE DO;
  IF FIRST.MONTHDATE THEN DO;
    AMTPAID = AMOUNT;
  END;
  /* SUM THE PAYMENTS ACROSS DUPLICATES (PAUL O'LEARY - AUG 2009) */
ELSE DO;
  AMTPAID = SUM(AMTPAID,AMOUNT);
END;
END;

IF LAST.MONTHDATE THEN OUTPUT;

RUN;

PROC PRINT DATA=&TYPE._PAYMENT (OBS=25); VAR SSN
  MONTHDATE PAY_NUM AMOUNT AMTPAID;; TITLE
  "AFTER FOR &TYPE DATA";
RUN;
```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```

DATA &TYPE._PAYMENT;
  SET &TYPE._PAYMENT;
  BY SSN;

/* PROCESS RECORDS TO RESOLVE PAYMENTS AND PRODUCE PERSON LEVEL DATA.
- RANGE OF MONTHS 2002-2010 */

/* SET UP MONTHLY ARRAYS FOR PERSON LEVEL VARIABLES */

LENGTH %DO I=1 %TO &TOT; &TYPE._TRIGGERED&&X&I %END; $10;
ARRAY PAYTYPE{*} $1 %DO I=1 %TO &TOT; &TYPE._TYPY&&X&I %END;;
ARRAY PAYAMT{*} %DO I=1 %TO &TOT; &TYPE._PYMT&&X&I %END;;
ARRAY TYPPAYNUM{*} $ %DO I=1 %TO &TOT; &TYPE._TYPNUM&&X&I %END;;
ARRAY DTPAY{*} %DO I=1 %TO &TOT; &TYPE._DTPY&&X&I %END;;
ARRAY PAYNUM{*} $ %DO I=1 %TO &TOT; &TYPE._PYNM&&X&I %END;;
ARRAY TRIG{*} $ %DO I=1 %TO &TOT; &TYPE._TRIGGERED&&X&I %END;;

/* SET UP ARRAYS FOR TEMPORARY VARIABLES */
ARRAY CYEAR (*) $ %DO I=1 %TO &TOT; C&&X&I %END;;
ARRAY NYEAR (*) %DO I=1 %TO &TOT; N&&X&I %END;;

RETAIN %DO I=1 %TO &TOT; &TYPE._TYPY&&X&I
&TYPE._TYPNUM&&X&I
&TYPE._PYMT&&X&I
&TYPE._DTPY&&X&I
&TYPE._PYNM&&X&I
&TYPE._TRIGGERED&&X&I

%END;;

/* INITIALIZE THE MONTHLY VARIABLES */
IF FIRST.SSN THEN DO;
  DO I = 1 TO &TOT;
    PAYTYPE(I) = ' ';
    TYPPAYNUM(I) = ' ';
    PAYAMT(I) = . ;
    DTPAY(I) = . ;
    PAYNUM(I) = ' ';
    TRIG(I) = ' ';
  END;
END;

/* THE FOLLOWING ROUTINE CREATES MONTHLY REFERENCE DATES FROM
JAN 2002 TO DEC 2011 */
J=1;
Y=&BEGYR;

DO I = 1 TO DIM(CYEAR); /* LOOP THROUGH ALL MONTH/YEAR REFERENCES */
  CYEAR(I) = PUT(Y,4.) || (PUT(J,Z2.));
  IF INT(I/12)=I/12 THEN DO; /*INCREMENT YEAR*/
    J=1;
    Y=Y+1;
  END;
END;

```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```
ELSE J=J+1; /*INCREMENT MONTH*/

NYEAR(I)=INPUT(SUBSTR(CYEAR(I),1,4)||SUBSTR(CYEAR(I),5,2)||'01',YYMMDD10.
);

IF MONTHDATE = NYEAR(I) THEN DO;
  PAYTYPE(I) = PAYMENT_TYPE;
  TYPPAYNUM(I) = TYPE_PAYMENT_NUMBER;
  PAYAMT(I) = AMTPAID;
  DTPAY(I) = DATE_PAYMENT;
  PAYNUM(I) = PAYMENT_NUMBER_SORT;
  TRIG(I) = TRIGGERED_BY;
END;
END;

DROP PYMT_TYPE PAYMENT_NUMBER_SORT AMOUNT AMTPAID
      MONTH MONTHDATE DATE_PAYMENT TRIGGERED_BY
      TYPE_PAYMENT_NUMBER AMOUNT EN DUNS
      MONTH_S YEAR PAY_NUM PAYMENT_TYPE I J Y;
DROP %DO I = 1 %TO &TOT; C&&X&I N&&X&I %END;;

LABEL
  %DO I=1 %TO &TOT;
    &TYPE._DTPY&&X&I = "&&X&I &TYPE Date Payment"
    &TYPE._TYPY&&X&I = "&&X&I &TYPE Type Payment"
    &TYPE._TYPNUM&&X&I = "&&X&I &TYPE Type Payment Number"
    &TYPE._PYMT&&X&I = "&&X&I &TYPE Payment Amount"
    &TYPE._PYNM&&X&I = "&&X&I &TYPE Payment Number"
    &TYPE._TRIGGERED&&X&I = "&&X&I &TYPE Triggered By"
  %END;;

IF LAST.SSN THEN OUTPUT;

RUN;

%MEND SUMPAY;

%SUMPAY (M);
%SUMPAY (O);

/* MERGE THE DATASETS AND THE EN INFORMATION AND OUTPUT FINAL FILE
*/

DATA OUT.ENPAYMENTS ;

RETAIN SSN EN_NAME: DUNS_ID;;

MERGE M_PAYMENT (IN=INM)
      O_PAYMENT (IN=INO)
      ENDATA;
BY SSN;
HAS_M_DATA = INM;
```

Appendix A.82
SAS Code: 2_PAYMENT_PERSON_RECORD

```
HAS_O_DATA = INO;

RUN;

TITLE 'FINAL PERSON LEVEL EN PAYMENTS DATA';
TITLE2;
PROC CONTENTS DATA=OUT.ENPAYMENTS ; RUN;
PROC FREQ; TABLES HAS_M_DATA*HAS_O_DATA/LIST MISSING; RUN;
PROC PRINT DATA=OUT.ENPAYMENTS (OBS=10) ; TITLE2 'FINAL OUTPUT
RECORDS' ;
RUN;

%MEND START;
%START;
```

Appendix A.83
JCL/SAS Code: RSACOMB

```
// #3590RSA JOB (12510000,T715,,SAS,,ITC9FL),3590BRON,
//           MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
// * PROJECT: Evaluation of the Ticket to Work Program-Part A
// *           Evaluation Implementation (6545) TRF09
// * PROJECT
// * DIRECTOR:      LAURA KOSAR
// *
// * PROGRAM:       OPDR.TG.PRD.ETTW.#3590.TRF10.RSA.PRDLIB(RSACOMB)
// *
// * DESCRIPTION:  COMBINED RSA SAS LOADED FILES
// *
// * DATE:          11/01/10 DAWN PHELPS (FERRAGAMO)
//*****
//RSA        EXEC SAS9,
//           WORK='120000,60000'
// *
// * INPUT
// * PRIOR TO 2002 THE SERVICES PROVIDED VARIABLES (ST PREFIX
VARIABLES)
// * WERE 1 POSITION BINARY (0/1) VARIABLES; IN 2002 THE FIELDS WERE
// * CHANGED TO 2 POSITIONS INDICATING BOTH THE VENDOR AND SOURCE OF
// * FUNDING. THE PROCESS OF COMBINING THE MULTIPLE YEARS TRUNCATED
THE
// * 2 POSITION FIELDS. TO CORRECT THIS WE ARE READING IN ALL THE
// * PREVIOUS SAS LOADED DATASETS AND SETTING THE LENGTH TO 2.
// *
// *RSAPREV DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF08P.RSAALL.SA.V2,DISP=SHR
//RSA98 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA98.SA.V1,DISP=SHR
//RSA99 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA99.SA.V1,DISP=SHR
//RSA00 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA00.SA.V1,DISP=SHR
//RSA01 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA01.SA.V1,DISP=SHR
//RSA02 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA02.SA.V1,DISP=SHR
//RSA03 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA03.SA.V1,DISP=SHR
//RSA04 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA04.SA.V1,DISP=SHR
//RSA05 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA05.SA.V1,DISP=SHR
//RSA06 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA06.SA.V1,DISP=SHR
//RSA07 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA07.SA.V1,DISP=SHR
//RSA08 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA08.SA.V1,DISP=SHR
//RSA09 DD DSN=OPDR.TG.PRD.ETTW.N4671.RSA09.SA.V1,DISP=SHR
//RSA10 DD DSN=OPDR.TG.PRD.ETTW.#2127.RSA10.SA.V1,DISP=SHR
// *
//RSA DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAALL.SA.V1,
//    DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//    SPACE=(CYL,(4000,2000),RLSE)
// *
//EVS DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAEVS.FL.V1,
//    DISP=(OLD,CATLG,DELETE),VOL=(,,10),
//    SPACE=(CYL,(3000,3000),RLSE)
// *
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAFIN.FL.V1,
//    DISP=(OLD,CATLG,DELETE),VOL=(,,10),
```

Appendix A.83
JCL/SAS Code: RSACOMB

```
//          SPACE=(CYL,(3000,300),RLSE)
//*
//TEMP1  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM98  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM99  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM00  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM01  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM02  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM03  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM04  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM05  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM06  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM07  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM08  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM09  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//TEM10  DD DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN  DD *
```

```
PROC FORMAT;
  VALUE $SEX '1'='M'
            '2'='F'
            ;
```

```
RUN;
```

```
*SORT THE DATA BY COSSN;
%MACRO START(YR = );
  PROC SORT DATA = RSA&YR..RSA&YR OUT = TEM&YR..RSA&YR;
    BY COSSN;
  %MEND START;
%START(YR = 98)
%START(YR = 99)
%START(YR = 00)
%START(YR = 01)
%START(YR = 02)
%START(YR = 03)
%START(YR = 04)
%START(YR = 05)
%START(YR = 06)
%START(YR = 07)
```

Appendix A.83
JCL/SAS Code: RSACOMB

```
%START(YR = 08)
%START(YR = 09)
%START(YR = 10)

DATA RSA.RSAALL;
  LENGTH ST: $2. AGENCY $3.
  STASSESS
  STDIAGTREAT
  STVOCREHAB
  STCOLLEGE
  STVOCTRAIN
  STOJT
  STBASICACDM
  STJOBREADY
  STAUGTRAIN
  STMISCTRAIN
  STJOBFIND
  STJOBPLACE
  STOJSUPP
  STTRANS
  STMAINTAIN
  STREHABTECH
  STREADER
  STINTERP
  STPERSATT
  STTECHASIST
  STINFO
  STOTR
  APPWRKSTAT $2.
  PRMYDIAG
  SCDYDIAG $4. ;
  SET /*RSAPREV.RSAALL(IN=RSAPREV)*/
    TEM98.RSA98(IN=RSA98)
    TEM99.RSA99(IN=RSA99)
    TEM00.RSA00(IN=RSA00)
    TEM01.RSA01(IN=RSA01)
    TEM02.RSA02(IN=RSA02)
    TEM03.RSA03(IN=RSA03)
    TEM04.RSA04(IN=RSA04)
    TEM05.RSA05(IN=RSA05)
    TEM06.RSA06(IN=RSA06)
    TEM07.RSA07(IN=RSA07)
    TEM08.RSA08(IN=RSA08)
    TEM09.RSA09(IN=RSA09)
    TEM10.RSA10(IN=RSA10)
    ;
  BY COSSN;

  * FIX ILLINOIS SEX;
  IF AGENCY IN ('12','5E','016','072')
    AND RSAYR IN ('02','03','04','05','06','07') THEN DO;
    IF RSASEX='1' THEN RSASEX='2';
    ELSE IF RSASEX='2' THEN RSASEX='1';
```

Appendix A.83
JCL/SAS Code: RSACOMB

```
END;

FILE EVS LRECL=107;
PUT @001 COSSN $CHAR9.
    @060 RSADOB YMMDDN8.
    @068 RSASEX $SEX.
    @107 ' '
    ;

IF LAST.COSSN THEN DO;
FILE FIN;
PUT @001 COSSN $CHAR9.
;
END;

RUN;

PROC CONTENTS DATA=RSA.RSAALL;
RUN;
```

Appendix A.84
JCL/SAS Code: READNUM

```

//#3590NUM JOB (12510000,T715,,SAS,,ITC9FL),BRONNIKOV-SVETLANA,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
//*
//*          *-----*
//*          |  OPDR.TG.PRD.ETTW.#3590.TRF10.RSA.PRDLIB(READNUM)  |
//*          *-----*
//*
//* LOAD RAW RSA NUMIDENT DATA INTO SAS
//* WRITTEN BY DAWN PHELPS
//* DATE:04/25/12 SVETLANA BRONNIKOV FOR TRF 10
//*
//* MPR PHONE 202 264-3460
//* E-MAIL SBRONNIKOV@MATHEMATICA-MPR.COM
//*****
//  SET REG='64M'
//*
//JS010 EXEC SAS9,
//      WORK='200000,100000',
//      REGION=&REG,
//      PARM='MEMSIZE=&REG'
//*
//IN1    DD DSN=OPDR.TG.PRD.ETTW.TRF10.RSANUM.R042012,DISP=SHR
//TEMP   DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//TEMP1  DD DSN=&&TEMP,
//      DISP=(NEW,DELETE,DELETE),
//      UNIT=TSILO
//OUT1   DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.RSANUM.SA.V1,
//      DISP=(OLD,CATLG,DELETE),
//      SPACE=(CYL,(1000,100),RLSE)
//*
OPTIONS NOCENTER COMPRESS=YES OBS=MAX;
/* READ IN NUMIDENT DATA */
DATA TEMP.RSANUM;
  INFILE IN1 TRUNCOVER;
  INPUT @027 REC $1.
  @;
/* READ DATA FROM DIFFERENT RECORD TYPES */
  IF REC IN ('G' 'J' 'K' '0' '1' '2' '4' '5' '9'
            'P' 'S' 'W' 'V') THEN DO;
    INPUT @13  SSN      $9.
           @27  ENTCDC $1.
           @29  CYCDTE $8.
           @37  FRSTNAM $15.
           @53  MIDNAM  $15.
           @69  LASTNAM  $20.
           @90  NAMESUF  $4.
           @97  DOBR     $8.
           @105 SEXR     $1.
           @106 RACER    $1.
  ;

```

Appendix A.84
JCL/SAS Code: READNUM

```
END;
ELSE IF REC IN ('D' 'L') THEN DO;
  INPUT @13 SSN      $9.
        @27 ENTCD   $1.
        @29 CYCDTE  $8.
        @37 FRSTNAM $15.
        @53 MIDNAM  $15.
        @69 LASTNAM $20.
        @90 NAMESUF $4.
        @97 DOBR    $8.
        @105 SEXR   $1.
        @298 DODR   $8.
      ;
END;
ELSE IF REC = 'T' THEN DO;
  INPUT @13 SSN      $9.
        @27 ENTCD   $1.
        @29 CYCDTE  $8.
        @37 FRSTNAM $15.
        @53 MIDNAM  $15.
        @69 LASTNAM $20.
        @90 NAMESUF $4.
        @97 DOBR    $8.
        @105 SEXR   $1.
        @145 DODR   $8.
      ;
END;
ELSE IF REC = 'E' THEN DO;
  INPUT @13 SSN      $9.
        @27 ENTCD   $1.
        @29 CYCDTE  $8.
        @37 FRSTNAM $15.
        @53 MIDNAM  $15.
        @69 LASTNAM $20.
        @90 NAMESUF $4.
        @97 DOBR    $8.
      ;
END;

IF SSN NE ' ';
RUN;
/* ORDER THE RECORDS CHRONOLOGICALLY */
PROC SORT DATA=TEMP.RSANUM OUT=TEMP1.RSANUM; BY SSN CYCDTE; RUN;
PROC PRINT DATA=TEMP1.RSANUM (OBS=100); TITLE 'AFTER SORT';
RUN;
DATA OUT1.RSANUM(KEEP=SSN
                ENTLST
                CYCDTLST
                DOB
                SEX
                RACE
                FRSTNAME
                MIDNAME
```

Appendix A.84
JCL/SAS Code: READNUM

```
                LASTNAME
                NAMESUFFIX
                DOD )
                ;
SET TEMP1.RSANUM; BY SSN CYCDTE;

/* DATA IS ORDERED BY CYCLE DATE FOR EACH SSN          */
/* READ THROUGH ALL RECORDS AND KEEP THE NON-BLANKS     */

LENGTH
    ENTLST    $1
    CYCDTLST  $8
    DOB       $8
    SEX       $1
    RACE      $1
    FRSTNAME  $15
    MIDNAME   $15
    LASTNAME  $20
    NAMESUFFIX $4
    DOD       $8
    ;
RETAIN
    ENTLST
    CYCDTLST
    DOB
    SEX
    RACE
    FRSTNAME
    MIDNAME
    LASTNAME
    NAMESUFFIX
    DOD
    ;
ARRAY ALL(*) ENTLST
            CYCDTLST
            DOB
            SEX
            RACE
            FRSTNAME
            MIDNAME
            LASTNAME
            NAMESUFFIX
            DOD;
IF FIRST.SSN THEN DO;
DO I = 1 TO DIM(ALL);
    ALL(I) = ' ';
END;
END;
IF DOBR NE ' ' THEN DOB = DOBR;
IF DODR NE ' ' THEN DOD = DODR;
IF SEXR NE ' ' THEN SEX = SEXR;
IF RACER GT '0' THEN RACE = RACER;
ENTLST = ENTCD;
```

Appendix A.84
JCL/SAS Code: READNUM

```
CYCDTLST = CYCDTE;  
IF FRSTNAM NE ' ' THEN FRSTNAME = FRSTNAM;  
IF MIDNAM NE ' ' THEN MIDNAME = MIDNAM;  
IF LASTNAM NE ' ' THEN LASTNAME = LASTNAM;  
IF NAMESUF NE ' ' THEN NAMESUFFIX = NAMESUF;  
IF LAST.SSN;  
RUN;  
PROC PRINT DATA=OUT1.RSANUM(OBS=50); TITLE 'AFTER LAST'; RUN;  
PROC FREQ DATA=OUT1.RSANUM; TABLES ENTLST; RUN;
```

Appendix A.85

JCL/SAS Code: EVS

```

//#3590EVS JOB (12510000,T715,,SAS,,ITC9FL),SVETLANA-BRONNIKOV,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
//* PROJECT: TRF 10
//*
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.#3590.TRF10.RSA.PRDLIB(EVS)
//*
//* DESCRIPTION: MPR-EVS SYSTEM -- VERIFIES SSNS BY COMPARING
//*                SSN, DOB, AND/OR GENDER ON A SUPPLIED FILE
//*                AGAINST THE RSA NUMIDENT FILE
//* WRITTEN BY DAWN PHELPS
//* DATE:       04/27/12 SVETLANA BRONNIKOV
//*****
//EVS      EXEC SAS9,
//          WORK='120000,60000'
//*
//* RSA NUMIDENT FILE
//IN1 DD   DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.RSANUM.SA.V1,DISP=SHR
//*
//* FILE TO BE VERIFIED
//IN2 DD   DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAEVS.FL.V1,DISP=SHR
//*
//OUT1 DD  DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAEVS.RP.V1,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE)
//OUT2 DD  DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAEVS.RSA.V1,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE)
//TEMP DD  DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,,10)
//SYSIN DD *

```

OPTIONS LS=132 COMPRESS=YES MPRINT OBS = MAX;

*

```

*
*   VERIFY CODE MAPPING
*
*   blank = initial value
*   S = no match in TRF.Demo
*   N = fails name check
*   G = fails gender check
*   Y = passes EVS
*   B = fails date of birth check
*   D = duplicate SSN
*   D = duplicate SSN, DOB, GENDER, NAME or duplicate SSN, DOB, NAME

```

(all

```

*   fields used for EVS are duplicates)

```

Appendix A.85
JCL/SAS Code: EVS

```

*
*
*****;

%LET NAME_CHK=0;      /*** 0=Don't perform name check / 1=Do ***/

/*****/
/** Macro to store number of observations in a given dataset ***/
/*****/
%MACRO NUMOBS(ds,mvar,condition);
  DATA TEMP.TEMP;
    SET &ds.&condition.;
  RUN;

  %global &mvar.;
  %let dsid = %sysfunc(open(TEMP.TEMP));
  %let &mvar. =%sysfunc(attrn(&dsid.,NOBS));
  %let rc = %sysfunc(close(&dsid));
%MEND;

/*****/
/**      Read in submitted data          ***/
/*****/
DATA TEMP.SUBMITTED_DATA;
  INFILE IN2 LRECL=107 PAD MISOVER JFCB=JFCB;

  INPUT SSN      $CHAR9.
        FNAME   $CHAR15.
        INIT    $CHAR15.
        LNAME   $CHAR20.
        DOB     YYMMDD8.
             GENDER $1.
             STATE $CHAR2.
             FILLER $CHAR37.
        ;
  FORMAT DOB MMDDYY10.;
  DOBYR=YEAR(DOB);

  * SAVE MACRO VARIABLE CONTAINING THE FILE TO BE VERIFIED;
  IF _N_=1 THEN
    CALL SYMPUT('SUBFILE',LEFT(TRIM(COMPRESS(SUBSTR(JFCB,1,44)))));

RUN;

/*****/
/** Merge with submitted data          ***/
/*****/
DATA TEMP.SUBMITTED_DATA_MATCH;
  MERGE TEMP.SUBMITTED_DATA(IN=IN1)
        IN1.RSANUM(IN=IN2
                  KEEP=SSN DOB SEX RENAME = (DOB = DOBRNUM)
                  );
  BY SSN;

```

Appendix A.85
JCL/SAS Code: EVS

```
IF IN1;
IF IN1 AND IN2 THEN MRG_SSN=1;
ELSE DO;
  MRG_SSN=0;
  VER_CODE="S";
END;

IF SEX = '1' THEN SEX = 'M';
ELSE IF SEX = '2' THEN SEX = 'F';
ELSE SEX = ' ';

***SCREEN OUT CASES WHICH HAVE THE WRONG DATA
FOR DOBRNUM IN NUMIDENT FILE***;

IF NOTDIGIT(TRIM(SUBSTR(DOBRNUM,5,4))) = 0
THEN NYR = SUBSTR(DOBRNUM,5,4);
NUMYR = INPUT(NYR,8.);

IF NOTDIGIT(TRIM(DOBRNUM)) = 0 THEN DO;
DOBNUM = INPUT(DOBRNUM,MMDDYY10.);
END;
FORMAT DOBNUM MMDDYY10.;
RUN;

DATA TEMP.SUBMITTED_DATA_MATCH;
SET TEMP.SUBMITTED_DATA_MATCH;
BY SSN;

NAME_CHK = &name_chk.;

DOB_SRCE=" ";
TRF_NAME=.;

/**** Proceed only if match during merge ****/
IF MRG_SSN=1 /*AND VFLAG=0*/ THEN DO;

  /**** Verify Name ****/
  IF &name_chk.=1 THEN DO;
    IF COMPRESS(BGN)=" " OR COMPRESS(BLN)=" " THEN DO;
      TRF_NAME=0;
      VER_CODE="N";
    END;
  ELSE DO;
    IF UPCASE(SUBSTR(LNAME,1,7)) NE UPCASE(SUBSTR(BLN,1,7))
    THEN DO;
      TRF_NAME=0;
      VER_CODE="N";
    END;
  ELSE DO;
    IF UPCASE(SUBSTR(FNAME,1,1)) NE UPCASE(SUBSTR(BGN,1,1))
    THEN DO;
      TRF_NAME=0;
      VER_CODE="N";
    END;
  END;
END;
```

Appendix A.85
JCL/SAS Code: EVS

```
        END;
    ELSE DO;
        IF COMPRESS(SUBSTR(INIT,1,1)) NE " "
        AND COMPRESS(SUBSTR(BMI,1,1)) NE " " THEN DO;
            IF UPCASE(SUBSTR(INIT,1,1))=UPCASE(SUBSTR(BMI,1,1))
            THEN TRF_NAME=1;
            ELSE DO;
                TRF_NAME=0;
                VER_CODE="N";
            END;
        END;
        ELSE DO;
            TRF_NAME=1;
        END;
    END;
END;
END;
END;
END;
END; /*** end name check ***/

/*** proceed on only if name checking was sucessful (or not done)*/
IF (&name_chk.=1 AND TRF_NAME=1) OR &name_chk.=0 THEN DO;

    /*** Verify DOB ***/
    IF /*MONTH(DOB)=MONTH(DOBMBR) AND*/ YEAR(DOB)=NUMYR
    AND NUMYR NE . THEN DOB_SRCE="R";
    ELSE VER_CODE="B";

    * IF RECORD PASSES DOB CHECK;
    IF DOB_SRCE = "R" THEN DO;
        /*** Verify Gender ***/

        * IF TRF HAS M,F GENDER THEN GO THRU VERIFICATION PROCESS;
        IF UPCASE(SEX) IN("M","F") THEN DO;
            IF UPCASE(GENDER)=UPCASE(SEX) THEN GENDER_VERIFY="Y";

            * IF INPUT DOES NOT HAVE M,F GENDER OR INPUT GENDER DOES NOT
            EQUAL TRF GENDER;
            ELSE DO;
                GENDER_VERIFY="N";
                VER_CODE="G";
            END;
        END;
    END;

    * IF TRF DOES NOT HAVE M,F GENDER BUT INPUT DOES THEN PASS;
    ELSE IF UPCASE(GENDER) IN ("M","F") THEN DO;
        GENDER_VERIFY="N";
        VER_CODE="Y";
    END;

    * IF TRF DOES NOT HAVE M,F GENDER AND INPUT DOES NOT HAVE M,F
    GENDER THEN FAIL;
    ELSE DO;
        GENDER_VERIFY="N";
```

Appendix A.85

JCL/SAS Code: EVS

```

        VER_CODE="G";
    END;
END;
/**** If case makes it this far then it passes verification ****/
IF DOB_SRCE NE " " AND GENDER_VERIFY="Y" THEN DO;
    VER_CODE="Y";
    VFLAG=1;
    END;
END;
END;
/**** end of MRG_SSN=1 ****/
RUN;

PROC FREQ;
    TABLES MRG_SSN*VER_CODE*TRF_NAME*DOB_SRCE*GENDER_VERIFY
    /MISSING LIST;
RUN;

/*****
***      Output File      ***
*****/

DATA OUT2.MPR_EVS4;
    SET TEMP.SUBMITTED_DATA_MATCH;
    KEEP /*STATUS*/ SSN FNAME INIT LNAME DOB GENDER STATE FILLER
        VER_CODE NAME_CHK MRG_SSN TRF_NAME DOB_SRCE
        DOBNUM SEX
    ;
RUN;

/*****
**      Output Report      ***
*****/
%NUMOBS(TEMP.SUBMITTED_DATA,LINE1,);
%NUMOBS(OUT2.MPR_EVS4,LINE5,);
%NUMOBS(OUT2.MPR_EVS4,LINE6,%STR( (WHERE=(VER_CODE='Y')) ));
%NUMOBS(OUT2.MPR_EVS4,LINE7,%STR( (WHERE=(VER_CODE NE 'Y')) ));
%NUMOBS(OUT2.MPR_EVS4,LINE8,%STR( (WHERE=(VER_CODE='S')) ));
%NUMOBS(OUT2.MPR_EVS4,LINE10,%STR( (WHERE=(VER_CODE='N')) ));
%NUMOBS(OUT2.MPR_EVS4,LINE11,%STR( (WHERE=(VER_CODE='B')) ));
%NUMOBS(OUT2.MPR_EVS4,LINE12,%STR( (WHERE=(VER_CODE='G')) ));

DATA _NULL_;
FILE OUT1;

PUT "MPR-EVS: Summary Results &SYSDATE.";
put "for Submitted File: &SUBFILE.";
IF &name_chk.=0 THEN PUT "Name checking was NOT performed";
ELSE PUT "Name checking WAS performed";
PUT;

PUT "&line1.=Total Input records";
PUT "&line5.=Output records";
PUT "&line6.=Input records that verified";

```

Appendix A.85
JCL/SAS Code: EVS

```
PUT "&line7.=Input records that did not verify";
PUT "&line8.=Input records that did not merge with TRF.DEMO";
PUT "&line10.=Input records that did not verify due to name";
PUT "&line11.=Input records that did not verify due to date of
birth";
PUT "&line12.=Input records that did not verify due to gender";

RUN;

PROC FREQ;
  TABLES VER_CODE /MISSING LIST;
RUN;
PROC PRINT DATA = OUT2.MPR_EVS4(OBS = 100);
WHERE DOB_SRCE = "R";
VAR DOB DOBNUM SEX;
RUN;
```

Appendix A.86
JCL/SAS Code: RSAEVS

```
//#3590EVS JOB (12510000,T715,,SAS,,ITC9FL),SVETLANA-BRONNIKOV,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
//* PROJECT: TRF (6545) TRF10
//*
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.#3590.TRF10.RSA.PRDLIB(RSAEVS)
//*
//* DESCRIPTION: MERGE EVS RESULTS TO ALL RSA YEARS
//* WRITTEN BY DAWN PHELPS
//* DATE:       05/01/12 SVETLANA BRONNIKOV
//*****
//EVS      EXEC SAS9,
//          WORK='120000,60000'
//*
//IN1 DD   DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAEVS.RSA.V1,DISP=SHR
//*
//RSA DD   DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAALL.SA.V1,DISP=SHR
//*
//OUT DD   DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAVERCD.SA.V1,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE)
//TEMP2 DD DSN=#3590.RSAALL.WEVS.SAS,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE)
//TEMP DD  DSN=&&TEMPX,UNIT=SYSDA,DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(4000,2000),RLSE),VOL=(,,10)
//SYSIN DD *

OPTIONS LS=132 COMPRESS=YES MPRINT MACROGEN OBS=MAX;

* Read in RSA EVS DATA;
DATA TEMP.EVS(DROP=RSADOB);/* DROP RASDOB IN ORDER TO KEEP ORIGINAL
*/

      * RENAME VARIABLE FOR MERGING AND DROP UNNECESSARY ONES;
      SET IN1.MPR_EVS4(RENAME=(DOB=RSADOB
                              GENDER=RSASEX
                              SSN=COSSN)
                      DROP=FNAME INIT LNAME STATE FILLER)
      ;
      * CREATE A DOB YEAR VARIABLE;
      DOBYR=YEAR(RSADOB);
      * CONVERT RSASEX FROM CHARACTER FLAGS TO NUMERIC FLAGS;
      IF RSASEX='M' THEN RSASEX='1';
      ELSE IF RSASEX='F' THEN RSASEX='2';

RUN;

* SORT RSA EVS DATA BY VERIFICATION KEY AND DESCENDING VER_CODE
* SO THAT THE VER_CODE='Y' WILL BE FIRST;
```

Appendix A.86
JCL/SAS Code: RSAEVS

```
PROC SORT DATA=TEMP.EVS;
  BY COSSN DOBYR RSASEX DESCENDING VER_CODE;
RUN;

* SORT RSA EVS DATA BY VERIFICATION KEY KEEPING ONLY THE FIRST
* UNIQUE OCCURENCE.  IF MULTIPILE OBS WITH THE KEY ARE FOUND
* AND ONE HAS A VER_CODE='Y' THEN THAT ONE WILL BE KEPT;
PROC SORT DATA=TEMP.EVS OUT=TEMP2.EVS NODUPKEYS;
  BY COSSN DOBYR RSASEX;
RUN;

TITLE1 'EVS';
PROC PRINT DATA=TEMP2.EVS(OBS=50);
RUN;

* READ RSA ALL YEARS SAS FILES;
DATA TEMP.RSAALL;
  SET RSA.RSAALL
    ;
  * CREATE A DOB YEAR VARIABLE;
  DOBYR=YEAR(RSADOB);
RUN;

PROC SORT DATA=TEMP.RSAALL OUT=TEMP2.RSAALL;
  BY COSSN DOBYR RSASEX;
RUN;

TITLE1 'RSA';
PROC PRINT DATA=TEMP2.RSAALL(OBS=50);
RUN;

* MERGE RSA EVS RESULTS WITH RSA ALL YEARS FILE;
DATA OUT.RSAVERCD;
  MERGE TEMP2.RSAALL(IN=RSA)
        TEMP2.EVS(IN=EVS)
    ;
  BY COSSN DOBYR RSASEX;

  * KEEP ALL RECORDS IN THE RSA ALL YEARS FILE;
  IF RSA;
  IF EVS THEN BOTH=1;
  ELSE BOTH=0;

RUN;

PROC FREQ DATA=OUT.RSAVERCD;
  TABLE VER_CODE;
  TABLE BOTH;
RUN;

TITLE1 'RSA W/ EVS';
PROC PRINT DATA=OUT.RSAVERCD(OBS=50);
RUN;
```

Appendix A.86
JCL/SAS Code: RSAEVS

ENDSAS ;

Appendix A.87
JCL/SAS Code: RSAFLAT

```
//#3590RSF JOB (12510000,T715,,SAS,,ITC9FL),SVETLANA-BRONNIKOV,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3590
//*****
//* PROJECT: TRF10
//*
//* PROJECT
//* DIRECTOR:   LAURA KOSAR
//*
//* PROGRAM:    OPDR.TG.PRD.ETTW.#3590.TRF10.RSA.PRDLIB(RSAFLAT)
//*
//* DESCRIPTION: FLATTEN RSA RECORDS TO ONE PER SSN
//*
//* DATE:       01/07/11 DAWN PHELPS
//* UPDATE:     05/01/12 SVETLANA BRONNIKOV
//*****
//*
//RSA      EXEC SAS9,
//          WORK='120000,60000'
//*
//* INPUT
//*
//IN       DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAVERCD.SA.V1,DISP=SHR
//*
//TEMP     DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.RSAFLAT.SA.V2,
//          DISP=(NEW,CATLG,DELETE),VOL=(,,10),
//          UNIT=TSILO
//SYSIN    DD *

OPTIONS OBS=MAX;

DATA TEMP.RSAFLAT(KEEP=COSSN RSAAGY: RSADOA: RSADOE: RSACLTYP:
                 RSADOC: RSASVS:);
    RETAIN RSAAGY1-RSAAGY20
           RSADOA1-RSADOA20
           RSADOE1-RSADOE20
           RSADOC1-RSADOC20
           RSASVS1-RSASVS20
           RSACLTYP1-RSACLTYP20
           RECCNT
           ;

    SET IN.RSAVERCD;

    * KEEP ONLY EVS VERIFIED RECORDS;
    WHERE VER_CODE='Y';
    BY COSSN;

    LENGTH RSADOA1-RSADOA20
           RSADOE1-RSADOE20
           RSADOC1-RSADOC20
```

Appendix A.87
JCL/SAS Code: RSAFLAT

```
RSASVS1-RSASVS20
  8.;
LENGTH RSAAGY1-RSAAGY20 $3.;
LENGTH RSACLTY1-RSACLTY20 $1.;

ARRAY RSAAGY RSAAGY1-RSAAGY20;
ARRAY RSADOA RSADOA1-RSADOA20;
ARRAY RSADOE RSADOE1-RSADOE20;
ARRAY RSADOC RSADOC1-RSADOC20;
ARRAY RSASVS RSASVS1-RSASVS20;
ARRAY RSACLTY RSACLTY1-RSACLTY20;

IF FIRST.COSSN THEN DO I=1 TO 20;
  RSAAGY(I)='  ';
  RSADOA(I)=.;
  RSADOE(I)=.;
  RSADOC(I)=.;
  RSASVS(I)=.;
  RSACLTY(I)='  ';
  RECCNT=1;
END;

RSAAGY(RECCNT)=AGENCY;
RSADOA(RECCNT)=APPDATE;
RSADOE(RECCNT)=RSADOE;
RSADOC(RECCNT)=CLOSEDATE;
RSASVS(RECCNT)=SVCS_PROV;
RSACLTY(RECCNT)=CLTYPE;

RECCNT=RECCNT+1;

IF LAST.COSSN THEN OUTPUT;

RUN;

%MACRO DOIT;
DATA OUT.RSAFLAT;
  SET TEMP.RSAFLAT(DROP = RSADOE);

  LABEL COSSN = "SOCIAL SECURITY NUMBER"
  %DO I = 1 %TO 20;
    RSAAGY&I = "RSA AGENCY &I"
    RSADOA&I = "RSA DATE OF APPLICATION &I"
    RSADOE&I = "RSA DATE OF ELIGIBILITY DETERMINATION &I"
    RSACLTY&I = "RSA CLOSURE TYPE &I"
    RSADOC&I = "RSA DATA OF CLOSURE &I"
    RSASVS&I = "RSA SERVICE USE FLAG &I"
  %END;
;

RUN;
%MEND DOIT;
%DOIT;
PROC PRINT DATA=IN.RSAVERCD(OBS=10);
```

Appendix A.87
JCL/SAS Code: RSAFLAT

```
WHERE VER_CODE='Y';  
RUN;  
  
PROC PRINT DATA=OUT.RSAFLAT(OBS=10);  
RUN;  
  
PROC CONTENTS DATA=OUT.RSAFLAT;  
RUN;  
  
ENDSAS;
```

Appendix A.88
JCL/SAS Code: LDWIND2

```

//$4671LDW JOB (12510000,T715,,SAS,,ITC9FL),DAWN-PHELPS,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*
//* *-----*
//* | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWIND2) |
//* *-----*
//* CREATE LDW INDICATOR FILE TO MERGE TO YEARLIES
//* REVISED FOR TRF08 TRF09
//* FOR DACS WE USE OPTION 2 - IF A DAC'S PRIMARY IS FLAGGED
//* ASSUME TERM AND SUSP INDICATORS ON THE DAC'S RECORD BELONG
//* TO THE PRIMARY. DO NOT FLAG THE DAC.
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* MODIFIED 12/21/2011 BY JEREMY PAGE FOR TRF10
//* MODIFIED 6/6/2012 BY DAWN PHELPS FOR TRF10 PATCH
//*****
//*
//JS010 EXEC SAS9,
// WORK='100000,50000'
//*
//DEM DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//IN2 DD
DD
DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T2WKDETN.SA.V1,DISP=SHR
//TEMP DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//TEMP1 DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//OUT DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10T.LDWDATA.SA.V2,
// DISP=(OLD,CATLG,DELETE),
// SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//*
//SYSIN DD *

/*****
*
* FILENAME: LDWIND.sas
* PROGRAMMER:MIRIAM LOEWENBERG
* PURPOSE:TO create revised DI MONTHLY LDW INDICATOR
* INPUT FILE COMBINED DEMO AND MBR DATA AND DCF DATA
* CREATED:6/29/10

*****
*/
options nocenter ls=132 ps=60 compress=binary MACROGEN MPRINT
OBS=MAX;
/* change endyr and endmn for each run */
%let begyr=1994;
%let endyr=2010; /* change as needed */
%let endmn=12; /* change as needed */

```

```

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

/* PUT "OUT" IN BELOW!!! */
DATA TEMP.LDWDATA (KEEP=SSN CAN BIC LDW: LAF: CDR:)

MATCHPRIM (KEEP=SSN CAN LDW:
            WORKFLAG TERMFLAG LAF: CDR:)
MATCHSEC (KEEP=SSN CAN WORKFLAG TERMFLAG LDW: LAF: CDR:
          RENAME=(WORKFLAG=SWORKFLAG TERMFLAG=STERMFLAG))
;
MERGE DEM.DMLDW
      (KEEP=SSN CAN DOBBEST DODBEST TWPCPLMNTH1
       IN=IND)
      IN1.MBR (IN=DW KEEP=SSN BIC RFD: WIC: CDR: RFST: LAF:)
      IN2.T2WKDETN (IN=E KEEP=COSSN TWPDATA: RENAME=(COSSN=SSN));
BY SSN;
IF IND AND DW;

/* correct bad values in twpcplmnth1 by reformatting */

IF TWPCPLMNTH1 GT '31DEC2011'D THEN
  TWPCPLMNTH1=MDY(SUBSTR(LEFT(TWPCPLMNTH1),5,2),01,
  SUBSTR(LEFT(TWPCPLMNTH1),1,4));

/* use dobest from demo to compute retiredate */

/* compute retirement date for comparison */
/* if b'day is january 1st - refer to previous year */
/* sas9 offers "sameday" which simplifies the code below */
/* basis for code is published table which gives retirement age
according to birth year */

IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN DO;
  YEARDOB = YEAR(DOBBEST) - 1;
  MONTHDOB = 12;
  DAYDOB = 31;
END;

ELSE DO;
  YEARDOB = YEAR(DOBBEST);

```

```

MONTHDOB = MONTH(DOBBEST);
DAYDOB = DAY(DOBBEST);
END;

```

```

BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
IF YEARDOB LE 1937 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (65*12), "SAMEDAY");
ELSE IF YEARDOB = 1938 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1939 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1940 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1941 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1942 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAMEDAY");
ELSE IF 1943 <= YEARDOB <= 1954 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (66*12), "SAMEDAY");
ELSE IF YEARDOB = 1955 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+2), "SAMEDAY");
ELSE IF YEARDOB = 1956 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+4), "SAMEDAY");
ELSE IF YEARDOB = 1957 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+6), "SAMEDAY");
ELSE IF YEARDOB = 1958 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+8), "SAMEDAY");
ELSE IF YEARDOB = 1959 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, ((66*12)+10), "SAMEDAY");
ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
  INTNX('MONTH', BIRTHDATE, (67*12), "SAMEDAY");

```

```

/* set up arrays for processing */

```

```

/* arrays for input */
ARRAY LAFI (*) %DO I = 1 %TO &TOT; LAF&&X&I %END;;
ARRAY WICI (*) %DO I = 1 %TO &TOT; WIC&&X&I %END;;
ARRAY RFDI (*) %DO I = 1 %TO &TOT; RFD&&X&I %END;;
ARRAY CDRI (*) %DO I = 1 %TO &TOT; CDR&&X&I %END;;
ARRAY RFSTI (*) %DO I = 1 %TO &TOT; RFST&&X&I %END;;
ARRAY TWPDT (*) %DO I = 1 %TO &TOT; TWPDATA&&X&I %END;;

```

```

/* for trf08 all records will be updated and rfst are on all
records - but it still may not improve the pre 2000 ldw's
appreciably */

```

```

/* begin with a check that the source values are before death
or retirement */

```

```

/* changes for trf07:
indicators set to missing if not on the rolls that month
twpdata helps to code work terminations */

```

```

/* changes from trf09 on:
look for a twp completion date as starting point for
evaluating ldw. first set the 0 for all months in pay status
after intializing all fields to 9. this is similar to the ssi
method.

```

```

        if the twpcmplmnth is missing look for the first triplet
        indicating suspension and use it as a starting point */

ARRAY LDW1{*}  %DO I=1 %TO &TOT; LDWDI&&X&I %END;;

/* find the month flagged by twpcmplmnth1 to begin processing.
   flag the death or retirement index to end processing */

ARRAY LDWDATE{*}  %DO I=1 %TO &TOT; LDWDATE&&X&I %END;;

LENGTH LDWNAME $9;
LDWNAME = ' ';
STOPIND = 0;
FOUNDWORK = 0;
DO I=1 TO DIM(LDW1);
/* construct date for the array element */
LDWNAME = VNAME(LDW1(I));
YR = SUBSTR(LDWNAME,6,2);
IF YR IN ('94','95','96','97','98','99') THEN YR = '19'||YR;
ELSE YR = '20'||YR;
YEAR = INPUT(YR,8.);
MONTH = INPUT(SUBSTR(LDWNAME,8,2),8.);
DAY = 1;
LDWDATE(I) = MDY(MONTH,DAY,YEAR);
END;

/* set the beginning month for processing */
DO I = 1 TO DIM(LDWDATE) UNTIL (FOUNDWORK GT 0);

IF TWPCMPLMNT1 NE . THEN DO;
/* if twp completion before 01/94 start at the beginning */
/* twpcmplmnth1 is always set to day 1 as is ldwdate */
IF TWPCMPLMNT1 LE LDWDATE(1) THEN FOUNDWORK = 1;
ELSE IF TWPCMPLMNT1 = LDWDATE(I) THEN FOUNDWORK = I;
END;

/* when missing twpcmplmnth then use 1st suspension triplet */
ELSE IF TWPCMPLMNT1 = . THEN DO;

/* starting at jan 1994 - look for a suspension triplet
   then start at that point when one is found */
DO I=1 TO DIM(LDWDATE) UNTIL (FOUNDWORK GT 0);
IF RFDI(I) = '7' AND
((WICI(I) = '2' OR WICI(I) = '7')
OR
(RFSTI(I) IN ('EPESGA' 'DIBWRK')))) THEN FOUNDWORK = I;
END; /* loop */

END; /* twpcmplmnth is missing */

END; /* foundwork gt 0 */

/* set the end month for processing */
DO I = 1 TO DIM(LDWDATE) UNTIL (STOPIND GT 0);
/* build indicators only up to retirement date or death date-
   missings begin in the month following the event */
IF (LDWDATE(I) GT RETIREDATE AND RETIREDATE GT .Z) OR
(LDWDATE(I) GT DODBEST AND DODBEST GT .Z) THEN STOPIND = I;

```

```

END;

/* where no work death or retirement limits processing set the
   stop index to the last month in the evaluation */
IF STOPIND = 0 THEN STOPIND = &TOT;

/* initialize the ldw variables */
DO I = 1 TO STOPIND;
  IF LAFI(I) = ' ' THEN LDW1(I) = .;
  ELSE IF SUBSTR(LAFI(I),1,1) IN ('C' 'E') THEN LDW1(I) = 0;
  ELSE IF LAFI(I) NE ' ' THEN LDW1(I) = 9;
END;

IF FOUNDWORK GT 0 THEN DO;

  DO I=FOUNDWORK TO STOPIND;
    /* work suspensions */
    IF RFDI(I) = '7' AND
      ((WICI(I) = '2' OR WICI(I) = '7')
       OR
        (RFSTI(I) IN ('EPESGA' 'DIBWRK'))
         THEN DO;
      LDW1(I) = 1;
      WORKFLAG = 1;
    END;
    /* work terminations - look at twpdatayymm in addition to cdr */
    ELSE IF RFDI(I) = 'T' AND WICI(I) IN ('8' '6') AND (CDRI(I) = 'E'
      OR TWPDT(I) = 'S')
      THEN DO;
      LDW1(I) = 2;
      TERMFLAG = 1;
    END;
  END; /*loop */

END; /* foundwork gt 0 */
DROP LDWDATE;;

* RECODE TERMINATIONS AFTER A SUSPENSION DUE TO WORK (LDWDI=1) AS A
  TERMINATION DUE TO WORK (LDWDI=2) PROVIDED THAT THE TERMINATION
  WAS NOT DUE TO A MEDICAL RECOVERY (CDRyyymm ne 'M'). THIS MAINLY
  EFFECTS BENES WHO COMPLETED A TWP AND ENTERED AN EPE;
DO I=2 TO STOPIND;
  IF LDW1(I)=9 AND LDW1(I-1) IN (1,2)
    AND SUBSTR(LAFI(I),1,1)='T'
    AND CDRI(I) NE 'M'
    THEN LDW1(I)=2;
END;

* ONCE TERMINATION FOR WORK ALWAYS TERMINATED FOR WORK UNTIL BACK
IN
  CURRENT PAY;
DO I=2 TO STOPIND;
  IF LDW1(I)=9 AND LDW1(I-1) IN (2,3) THEN LDW1(I)=3;
  ELSE IF LDW1(I) NOT IN (0,.) AND LDW1(I-1) IN (2,3)
    THEN LDW1(I)=LDW1(I-1);
END;

/* output 3 files - 1. flagged primaries 2. flagged dacs

```

3. unflagged cases.

we do this because we need to determine whether work data on
a dac record belongs to the primary and not the dac */

```
IF WORKFLAG = 1 OR TERMFLAG = 1 THEN DO;
  IF SUBSTR(BIC,1,1) = 'C' THEN OUTPUT MATCHSEC; /* dacs */
  ELSE OUTPUT MATCHPRIM; /* primaries */
END;
ELSE OUTPUT TEMP.LDWDATA; /*no work indication on these records*/
```

*****This section is used if we have a merge statement issue again;

```
* IF WORKFLAG = 1 OR TERMFLAG = 1 THEN DO;
*   IF SUBSTR(BIC,1,1) = 'C' THEN OUTPUT MATCHSEC; /* dacs */
**  ELSE IF BIC='A' THEN OUTPUT MATCHPRIM; /* primaries */
*   ELSE OUTPUT TEMP.LDWDATA; /* not primaries or dacs */
* END;
*   ELSE OUTPUT TEMP.LDWDATA; /* not primaries or dacs */
```

RUN;

/* check the number of dac records where the ssn = can. this may be
the case for old records where the rule to always put the
dependent

boan on the record was not yet enforced. in the old days if the
bene's own ssn was not known (a dependent bene) the ssn of the
primary was entered into the boan field */

```
PROC FREQ DATA=TEMP.LDWDATA;
  TABLES BIC; WHERE SUBSTR(BIC,1,1) = 'C' AND CAN = SSN;
  TITLE 'CAN = SSN AND BIC = C';
RUN;
```

```
PROC SORT DATA=MATCHPRIM; BY CAN; RUN;
PROC FREQ; TABLES WORKFLAG*TERMFLAG/LIST MISSING;
  TITLE 'PRIMARY CASES WITH WORK INDICATED';
RUN;
```

```
PROC SORT DATA=MATCHSEC; BY CAN; RUN;
PROC FREQ; TABLES SWORKFLAG*STERMFLAG/LIST MISSING;
  TITLE 'DAC CASES WITH WORK INDICATED';
RUN;
```

```
DATA MATCHSEC (KEEP=SSN LDW: );
  MERGE MATCHPRIM (IN=P DROP=SSN LDW:)
  MATCHSEC (IN=S); BY CAN;
```

/* keep only dac cases */

```
ARRAY LDW{*} %DO I=1 %TO &TOT; LDWDI&&X&I %END;;
```

/* if dac has a primary who has been flagged - reset indicator
to 9 for the dac because the work indication most likely
belongs to the primary */

```
IF P AND S AND CAN NE ' ' THEN DO;
  IF WORKFLAG = 1 AND SWORKFLAG = 1 THEN DO;
    DO I = 1 TO DIM(LDW);
```

```

        IF LDW(I) = 1 THEN LDW(I) = 9;
    END;
END;
IF TERMFLAG = 1 AND STERMFLAG = 1 THEN DO;
    DO I = 1 TO DIM(LDW);
        IF LDW(I) = 2 THEN LDW(I) = 9;
    END;
END;
END;

IF S;

RUN;

PROC PRINT DATA=MATCHSEC (OBS=25); TITLE 'REVISED FLAGGED DAC
RECORDS';
RUN;
PROC PRINT DATA=MATCHPRIM (OBS=25); TITLE 'FLAGGED PRIMARY RECORDS';
RUN;

/* put the pieces back together */
PROC SORT DATA= MATCHPRIM; BY SSN; RUN;
PROC SORT DATA= MATCHSEC; BY SSN; RUN;
DATA TEMP1.LDWDATA (KEEP=SSN LDW: );
SET TEMP.LDWDATA MATCHPRIM MATCHSEC;
BY SSN;
RUN;
/* dedup the few duplicates because of bad boans */
PROC SORT DATA=TEMP1.LDWDATA OUT=OUT.LDWDATA NODUPKEY; BY SSN; RUN;
PROC FREQ DATA=OUT.LDWDATA;
TABLES LDW:/ MISSPRINT;
RUN;
PROC PRINT DATA=OUT.LDWDATA (OBS=25); RUN;

%MEND;

%START;

```

Appendix A.89
JCL/SAS Code: LDWSSI

```

//$4671T16 JOB (12510000,T715,,SAS,,ITC9FL),DAWN-PHELPS,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*
//*      *-----*
//*      | OPDR.TG.PRD.ETTW.N4671.TRF10.LDW.PRDLIB(LDWSSI) |
//*      *-----*
//*
//* CREATE REVISED SSI INDICATORS
//* REVISION FOR TRF10
//* CONTACT MIRIAM LOEWENBERG
//* SSA PHONE 202 358-6214 MPR PHONE 202 484-4829
//* E-MAIL MLOEWENBERG@MATHEMATICA-MPR.COM
//* CONSTRUCT SSI LEFT DUE TO WORK INDICATORS
//* MODIFIED      12/21/2011 JEREMY PAGE FOR TRF10
//*****
//*
//JS010    EXEC SAS9,
//          WORK='120000,60000'
//*
//TEMP1    DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//DEM      DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
//IN1      DD DSN=OPDR.TG.PRD.ETTW.N8043.T16MRG31.Y2010.SSD,DISP=SHR
//IN2      DD DSN=OPDR.TG.PRD.ETTW.$2358.TRF10P.T16EARN.SA.V1,DISP=SHR
//OUT      DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10T.LDWSSI10.SSD,
//          DISP=(OLD,CATLG,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,10)
//*        DISP=(NEW,CATLG,DELETE),
//*        UNIT=TSILO
//SYSIN    DD *
OPTIONS NOCENTER COMPRESS=BINARY OBS=MAX;
%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;

```

```

%let tot=%eval(&k-1);

DATA OUT.LDWSSI (KEEP=SSN LDWSSI:);
MERGE DEM.DMLDW
      (IN=D KEEP=SSN DOBBEST DODBEST)
IN1.LONG (IN=L KEEP=SSN PST: EIN: UIN: MFT)
IN2.T16EARN (RENAME=(COSSN=SSN));
  BY SSN;
  IF L ;
/* use dobbest from demo to compute retiredate */

/* compute retirement date for comparison */
/* if b'day is january 1st - refer to previous year */
/* sas9 offers "sameday" which simplifies the code below */
/* basis for code is published table which gives retirement age
   according to birth year */

IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN DO;
  YEARDOB = YEAR(DOBBEST) - 1;
  MONTHDOB = 12;
  DAYDOB = 31;
END;

ELSE DO;
  YEARDOB = YEAR(DOBBEST);
  MONTHDOB = MONTH(DOBBEST);
  DAYDOB = DAY(DOBBEST);
END;

  BIRTHDATE = MDY(MONTHDOB, DAYDOB, YEARDOB);
  IF YEARDOB LE 1937 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, (65*12), "SAMEDAY");
  ELSE IF YEARDOB = 1938 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+2), "SAMEDAY");
  ELSE IF YEARDOB = 1939 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+4), "SAMEDAY");
  ELSE IF YEARDOB = 1940 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+6), "SAMEDAY");
  ELSE IF YEARDOB = 1941 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+8), "SAMEDAY");
  ELSE IF YEARDOB = 1942 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((65*12)+10), "SAMEDAY");
  ELSE IF 1943 <= YEARDOB <= 1954 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, (66*12), "SAMEDAY");
  ELSE IF YEARDOB = 1955 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((66*12)+2), "SAMEDAY");
  ELSE IF YEARDOB = 1956 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((66*12)+4), "SAMEDAY");
  ELSE IF YEARDOB = 1957 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((66*12)+6), "SAMEDAY");
  ELSE IF YEARDOB = 1958 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((66*12)+8), "SAMEDAY");
  ELSE IF YEARDOB = 1959 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, ((66*12)+10), "SAMEDAY");
  ELSE IF YEARDOB GE 1960 THEN RETIREDATE =
    INTNX('MONTH', BIRTHDATE, (67*12), "SAMEDAY");

/* create ssi ldw indicators */

```

```

/* input arrays - psta vars are renamed for the annual files */
ARRAY PSTA(*) %DO I = 1 %TO &TOT; PST&&X&I %END;;
ARRAY T16GRS(*) %DO I = 1 %TO &TOT; T16GRSAMT&&X&I %END;;
ARRAY T16NET(*) %DO I = 1 %TO &TOT; T16NETAMT&&X&I %END;;
ARRAY T16EXL(*) %DO I = 1 %TO &TOT; T16EXLAMT&&X&I %END;;
ARRAY T16EXP(*) %DO I = 1 %TO &TOT; T16EXPAMT&&X&I %END;;
ARRAY T16PAS(*) %DO I = 1 %TO &TOT; T16PASAMT&&X&I %END;;
ARRAY EICM(*) %DO I = 1 %TO &TOT; EIN&&X&I %END;;
ARRAY UICM(*) %DO I = 1 %TO &TOT; UIN&&X&I %END;;

/* NEW ARRAYS */
ARRAY SGALEV(*) %DO I = 1 %TO &TOT; SGALEVEL&&X&I %END;;
ARRAY LDWDATE(*) %DO I = 1 %TO &TOT; LDWDATE&&X&I %END;;
ARRAY DCFEARN(*) %DO I = 1 %TO &TOT; DCFEARN&&X&I %END;;
ARRAY SSREARN(*) %DO I = 1 %TO &TOT; SSREARN&&X&I %END;;
ARRAY SSILDW(*) %DO I = 1 %TO &TOT; LDWSSI&&X&I %END;;

/* construct an array of dates corresponding to the month/year
fields in the source arrays for LDW construction */

LENGTH LDWNAME $11;

LDWNAME = ' ';
DO I=1 TO DIM(SSILDW);
/* construct date for the array element */
LDWNAME = VNAME(SSILDW(I));
YR = SUBSTR(LDWNAME,7,2);
IF YR IN (94,95,96,97,98,99) THEN YR = '19' || YR;
ELSE YR = '20' || YR;
YEAR = INPUT(YR,8.);
MONTH = INPUT(SUBSTR(LDWNAME,9,2),8.);
DAY = 1;
LDWDATE(I) = MDY(MONTH,DAY,YEAR);
END;

/* set sga levels for time - use LDWDATEeyymm to target the
year/month time frame. sga levels are higher for blind */

%MACRO SGAIT (DATE1,DATE2,LEV1,LEV2);
DO I = 1 TO DIM(LDWDATE);

IF LDWDATE(I) GE &DATE1 AND LDWDATE(I) LE &DATE2 THEN DO;
IF SUBSTR(MFT,1,1) = 'B' THEN SGALEV(I) = &LEV1;
ELSE SGALEV(I) = &LEV2;
END;

END;

%MEND;

%SGAIT('01JAN1994'D,'31DEC1994'D,930,500);
%SGAIT('01JAN1995'D,'31DEC1995'D,940,500);
%SGAIT('01JAN1996'D,'31DEC1996'D,960,500);
%SGAIT('01JAN1997'D,'31DEC1997'D,1000,500);
%SGAIT('01JAN1998'D,'31DEC1998'D,1050,500);
/* 1999 has 2 levels */
%SGAIT('01JAN1999'D,'30JUN1999'D,1110,500);

```

```

%SGAIT('01JUL1999'D,'31DEC1999'D,1110,700);
%SGAIT('01JAN2000'D,'31DEC2000'D,1170,700);
%SGAIT('01JAN2001'D,'31DEC2001'D,1240,740);
%SGAIT('01JAN2002'D,'31DEC2002'D,1300,780);
%SGAIT('01JAN2003'D,'31DEC2003'D,1330,800);
%SGAIT('01JAN2004'D,'31DEC2004'D,1350,810);
%SGAIT('01JAN2005'D,'31DEC2005'D,1380,830);
%SGAIT('01JAN2006'D,'31DEC2006'D,1450,860);
%SGAIT('01JAN2007'D,'31DEC2007'D,1500,900);
%SGAIT('01JAN2008'D,'31DEC2008'D,1570,940);
%SGAIT('01JAN2009'D,'31DEC2009'D,1640,980);
%SGAIT('01JAN2010'D,'31DEC2010'D,1640,1000);

/* create earnings fields from eincm variables using formula.
   they will represent earnings from 1994 through march 2001 */

/* intialize ssi earnings to 0 */

DO I = 1 TO DIM(SSREARN);

   SSREARN(I) = 0;

END;

DO I = 1 TO DIM(SSREARN);

   IF EICM(I) GT 0 AND UICM(I) GT 0 THEN DO;

      SSREARN(I) = (EICM(I)*2) + 65;

   END;

   ELSE IF EICM(I) GT 0 THEN DO;

      SSREARN(I) = (EICM(I)*2) + 85;

   END;

END;

/* create earnings amounts using dcf earnings fields
   and exclusion amounts */

DO I = 1 TO DIM(DCFEARN);
/* sum gross amounts and net amounts - these are wages and
   self-employment earnings. In most cases only one or the
   other will have values */
SUMEARN = SUM(OF T16GRS(I),T16NET(I),0);
/* sum the exclusion amounts */
SUMEXCEP = SUM(OF T16EXL(I),T16PAS(I),T16EXP(I),0);
/* subtract the exclusion amounts - the result are earnings
   to be evaluated with reference to sga */
DCFEARN(I) = SUMEARN-SUMEXCEP;

END;

/* intialize ssi ldw to 9 - reset to 0 if c01- to 9 if dead */

```

```

DO I = 1 TO DIM(PSTA);

  IF PSTA(I) GT ' ' THEN SSILDW(I) = 9;
  IF PSTA(I) = 'C01' THEN SSILDW(I) = 0;
  ELSE IF PSTA(I) = 'T01' THEN SSILDW(I) = 9;

END;

/* find first occurrence of work - psta = n01 and earnings
   gt 0. up until april 2001 use the ssr eincm var.
   from then on use the dcf earnings */

FOUNDWORK = 0;

DO I = 1 TO DIM(PSTA) UNTIL (FOUNDWORK GT 0);

  IF LDWDATE(I) GE '01JAN1994'D AND LDWDATE(I) LE '31MAR2001'D THEN
DO;
  IF PSTA(I) = 'N01' AND SSREARN(I) GT 0 THEN FOUNDWORK = I;
  END;
  ELSE IF PSTA(I) = 'N01' AND DCFEARN(I) GT 0 THEN FOUNDWORK = I;
  END;

/* evaluate work with change in status - loop thru all the changes -
   process all months indicating work -
   all other codes have been set already */

IF FOUNDWORK GT 0 THEN DO;

  DO I = FOUNDWORK TO DIM(PSTA);

    IF PSTA(I) = 'N01' THEN SSILDW(I) = 1;

    ELSE IF PSTA(I) IN ('N04' 'N06') THEN DO;
    /* check if earnings are at or above sga */

      IF LDWDATE(I) GE '01JAN1994'D AND LDWDATE(I) LE '31MAR2001'D
      THEN DO;
      IF SSREARN(I) GE SGALEV(I) THEN SSILDW(I) = 1;
      END;
      ELSE IF DCFEARN(I) GE SGALEV(I) THEN SSILDW(I) = 1;
      END;

    /* code work terminations except for death */
    ELSE IF SUBSTR(PSTA(I),1,1) = 'T' AND PSTA(I) NE 'T01'
    THEN SSILDW(I) = 2;

  END;

END;

* ONCE A BENE IS TERMINATED FOR WORK HE REMAINS TERMINATED FOR WORK
  UNTIL A MONTH OF CURRENT PAY;
DO I=2 TO DIM(SSILDW);
  IF SSILDW(I-1)=2 AND SSILDW(I) NE 0 THEN SSILDW(I)=2;
END;

```

```

* ONCE A BENE IS IN SUSPENDED FOR WORK STATUS FOR 12 MONTHS ALL
  SUBSEQUENT MONTHS ARE MARKED AS LDW=3 UNTIL A MONTH OF CURRENT PAY
  OR ACTUAL TERMINATION;
DO I=1 TO DIM(SSILDW);
  IF SSILDW(I)=1 THEN COUNT1S=SUM(COUNT1S,1);
  IF SSILDW(I) NE 1 THEN COUNT1S=0;
  IF COUNT1S=13 THEN DO J=I TO DIM(SSILDW)
    WHILE(SSILDW(J) NOT IN (0,2));
    SSILDW(J)=3;
  END;
END;

/* set to missing if dead or retired - pstat codes of t01 which
  indicates death have been set to 9 above - overwrite to missing
  using the DODBEST date from the DEMO file - the best indication
  of death in the TRF. If there is not a concurrence in the t01
month
  and the month of the death date then leave the LDW set to 9 */

DO I = 1 TO DIM(SSILDW);

  IF (LDWDATE(I) GT RETIREDATE AND RETIREDATE GT .Z) OR
    (LDWDATE(I) GT DODBEST AND DODBEST GT .Z) THEN SSILDW(I) = .;

END;

LABEL
%DO I=1 %TO &TOT;
LDWSSI&&X&I = "&&X&I SSI LEFT DUE TO WORK INDICATOR"
%END;
;
RUN;

%MEND;
%START;

PROC CONTENTS DATA=OUT.LDWSSI;
TITLE "CONTENTS OF LDWSSI";
RUN;

PROC FREQ DATA=OUT.LDWSSI;
TABLES LDWSSI:/MISSPRINT;
RUN;

```

Appendix A.90
JCL/SAS Code: LDWCOMB

```

//$4671LDC JOB (12510000,T715,,SAS,,ITC9FL),DAWN-PHELPS,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.N4671.TRF10.LDW.PRDLIB(LDWCOMB) |
//*          *-----*
//*
/** CREATE REVISED COMBINED LDW INDICATORS FOR TRF10
/** CONSTRUCT COMBINED LEFT DUE TO WORK INDICATORS
/** MODIFIED 12/23/2011 BY JEREMY PAGE
/** MODIFIED 06/13/2012 BY DAWN PHELPS
//*****
/**
//JS010 EXEC SAS9,
//          WORK='120000,60000'
/**
//TEMP1 DD DSN=&&TEMP,
//          DISP=(NEW,DELETE,DELETE),
//          SPACE=(CYL,(1000,100),RLSE),VOL=(,,8)
//IN1 DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10T.LDWSSI10.SSD,DISP=SHR
//IN2 DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10T.LDWDATA.SA.V2,DISP=SHR
//IN3 DD DSN=OPDR.TG.PRD.ETTW.N8043.MRGPBEN.D1012.SSD,DISP=SHR
//FIN DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10.LINKSSN.SSD,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWCOMB.SA.V2,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD *
OPTIONS NOCENTER COMPRESS=BINARY OBS=MAX;

%LET BEGYR=1994;
%LET ENDYR=2010; /* CHANGE AS NEEDED */
%let endmn=12; /* change as needed */

/* step to assign macro variables to handle time series data */

%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

DATA OUT.LDWCOMB (KEEP=SSN LDWSSI LDWDI LDWCM:);

```

```

MERGE IN1.LDWSSI
      ( KEEP=SSN
      %DO I=1 %TO &TOT;
          LDWSSI&&X&I
      %END; )
      IN2.LDWDATA
      ( KEEP=SSN
      %DO I=1 %TO &TOT;
          LDWDI&&X&I
      %END; )
      IN3.MBR
      ( KEEP=SSN
      %DO I=1 %TO &TOT;
          RFST&&X&I
      %END; )
      FIN.LINKSSN
      ( KEEP=SSN IN=L9)
      ;
BY SSN;
IF L9;

/* new 2008 - create a program-combined ldw indicator */

ARRAY NEWLDW (*) %DO I=1 %TO &TOT; LDWCM&&X&I %END;;
ARRAY DILDW  (*) %DO I=1 %TO &TOT; LDWDI&&X&I %END;;
ARRAY SSILDW (*) %DO I=1 %TO &TOT; LDWSSI&&X&I %END;;
ARRAY RFST   (*) %DO I=1 %TO &TOT; RFST&&X&I %END;;

DO I = 1 TO DIM(NEWLDW);
  /* if not in one pgm combined ldw=ldw from other pgm */
  IF SSILDW(I)=. THEN NEWLDW(I)=DILDW(I);
  ELSE IF DILDW(I)=. THEN NEWLDW(I)=SSILDW(I);
  /* if in current pay status in either pgm combined ldw=0 */
  ELSE IF DILDW(I)=0 OR SSILDW(I)=0 THEN NEWLDW(I)=0;
  /* new for trf09 check prison suspension on di side */
  /* if suspended due to work in either pgm combined ldw=1 */
  ELSE IF DILDW(I)=1 OR (SSILDW(I)=1 AND RFST(I) NE 'PRISON')
    THEN NEWLDW(I) = 1;
  /* if terminated due to work in either pgm combine ldw=2 */
  ELSE IF DILDW(I)=2 OR (SSILDW(I)=2 AND RFST(I) NE 'PRISON')
    THEN NEWLDW(I)=2;
  ELSE IF DILDW(I)=3 OR (SSILDW(I)=3 AND RFST(I) NE 'PRISON')
    THEN NEWLDW(I)=3;
  /* if inelig for other reason in both pgms combine ldw=9 */
  ELSE NEWLDW(I)=9;
END;

RUN;
%MEND;
%START;

PROC CONTENTS DATA=OUT.LDWCMB;
TITLE "CONTENTS OF LDWCMB";
RUN;

PROC PRINT DATA=OUT.LDWCMB (OBS=25); RUN;
PROC FREQ DATA=OUT.LDWCMB;
TABLES LDW:/ MISSING;

```

RUN;

Appendix A.91
JCL/SAS Code: LDWSTRNG

```
//$4671LDC JOB (12510000,T715,,SAS,,ITC9FL),DAWN-PHELPS,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=$4671
//*****
//*          *-----*
//*          | OPDR.TG.PRD.ETTW.#2127.TRF10.LDW.PRDLIB(LDWSTRNG) |
//*          *-----*
//*
//* PURPOSE: STRING OUT 3'S IN THE LDW INDICATORS
//* CONTACT JEREMY PAGE
//* SSA PHONE 202 358-6228 MPR PHONE 202 554-7515
//* E-MAIL JPAGE@MATHEMATICA-MPR.COM
//*
//* CREATED: 02/02/2011 BY JEREMY PAGE
//* MODIFIED: 06/13/2012 BY DAWN PHELPS
//*****
//*
//JS010 EXEC SAS9,
//          WORK='120000,60000'
//*
//IN DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWCOMB.SA.V2,DISP=SHR
//DEM DD DSN=OPDR.TG.PRD.ETTW.#3590.TRF10P.DMLDW.SA.V1,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,
//          DISP=(OLD,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD *
OPTIONS NOCENTER MACROGEN COMPRESS=BINARY OBS=MAX;

%MACRO DOIT;

DATA OUT.LDWSTRNG (KEEP=SSN LDWSSI: LDWDI: LDWCM:);
MERGE IN.LDWCOMB
      DEM.DMLDW (KEEP=SSN DOBBEST DODBEST);
BY SSN;

***CALCULATE FRA;
***WHEN CALCULATING FRA PEOPLE BORN ON JAN 1 ARE CONSIDERED
***TO HAVE A BIRTH DATE OF DEC 31 OF THE PREVIOUS YEAR.;
IF MONTH(DOBBEST) = 1 AND DAY(DOBBEST) = 1 THEN
    RETIRE_DOB=DOBBEST-1;
ELSE RETIRE_DOB=DOBBEST;

IF YEAR(RETIRE_DOB) LE 1937 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,(65*12),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1938 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+2),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1939 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+4),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1940 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+6),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1941 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+8),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1942 THEN RETIREDATE =
    INTNX('MONTH',RETIRE_DOB,((65*12)+10),"SAMEDAY");
```

```

ELSE IF 1943 <= YEAR(RETIRE_DOB) <= 1954 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,(66*12),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1955 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,((66*12)+2),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1956 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,((66*12)+4),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1957 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,((66*12)+6),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1958 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,((66*12)+8),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) = 1959 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,((66*12)+10),"SAMEDAY");
ELSE IF YEAR(RETIRE_DOB) GE 1960 THEN RETIREDATE =
  INTNX('MONTH',RETIRE_DOB,(67*12),"SAMEDAY");

*****
***CONSTRUCT A DATE THAT IS THE EARLIEST OF FRA
***DOD OR DEC 31, 2010. THIS WILL ALLOW THE LOOP TO
***TO STOP STRINGING 3 WHEN A PERSON IS DEAD OR FRA.
*****;
ENDDATE=MIN(RETIREDATE,DODBEST,'31DEC2010'D);

STOPMNTN=INTCK('MONTH','01JAN1994'D,ENDDATE)+1;

***ARRAY REFERENCES FOR LDW VARIABLES;
ARRAY SSI_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWSSI&YR.01-LDWSSI&YR.12
                %END;;

ARRAY DI_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWDI&YR.01-LDWDI&YR.12
                %END;;

ARRAY COMB_ARRAY (*) %DO YEAR = 1994 %TO 2010;
                %LET YR=%SUBSTR(&YEAR.,3,2);
                LDWCM&YR.01-LDWCM&YR.12
                %END;;

***STRING 3 IN LDW VARIABLES;
DO I = 2 TO STOPMNTN;
  ***STRING LDWSSI;
  IF SSI_ARRAY(I) = . AND SSI_ARRAY(I-1) IN (2,3)
    THEN SSI_ARRAY(I) = 3;

  ***STRING LDWDI;
  IF DI_ARRAY(I) = . AND DI_ARRAY(I-1) IN (2,3)
    THEN DI_ARRAY(I) = 3;

  ***STRING LDWCOMB;
  IF COMB_ARRAY(I) = . AND COMB_ARRAY(I-1) IN (2,3)
    THEN COMB_ARRAY(I) = 3;
END;

PROC CONTENTS DATA=OUT.LDWSTRNG;
RUN;

```

```
%MEND DOIT;  
%DOIT;
```

Appendix A.92
JCL: LDW94JCL - LDW10JCL

```
//#3836Y94 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836  
//*****  
//*  
//*          *-----*  
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW94JCL)|  
//*          *-----*  
//*  
//* JCL FOR YEAR 1994 TO RUN LDWPATCH  
//*  
//* CREATED BY: PATSY LLOYD 7/13/12  
//* MPR PHONE 202 484-4240  
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM  
//*  
//*****  
//*  
//JS094 EXEC SAS9,  
//      WORK='100000,50000',  
//      PARM='SYSPARM="1994" '  
//*  
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1994.SA.V2,DISP=SHR  
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR  
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1994.SA.V3,  
//      DISP=(NEW,CATLG,DELETE),  
//      UNIT=TSILO  
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),  
//      DISP=(SHR,PASS,KEEP)  
//*
```

```
//#3836Y95 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,  
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836  
//*****  
//*  
//*          *-----*  
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW95JCL)|  
//*          *-----*  
//*  
//* JCL FOR YEAR 1995 TO RUN LDWPATCH  
//*  
//* CREATED BY: PATSY LLOYD 7/13/12  
//* MPR PHONE 202 484-4240  
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM  
//*  
//*****  
//*  
//JS095 EXEC SAS9,  
//      WORK='100000,50000',  
//      PARM='SYSPARM="1995" '  
//*  
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1995.SA.V2,DISP=SHR  
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR  
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1995.SA.V3,
```

```

//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//          DISP=(SHR,PASS,KEEP)
//*

//#3836Y96 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW96JCL)|
//*          *-----*
//*
//* JCL FOR YEAR 1997 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//*
//JS096   EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="1996"'
//*
//OLD    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1996.SA.V2,DISP=SHR
//LDW    DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT    DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1996.SA.V3,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//          DISP=(SHR,PASS,KEEP)
//*

//#3836Y97 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW97JCL)|
//*          *-----*
//*
//* JCL FOR YEAR 1997 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//*
//JS097   EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="1997"'
//*
//OLD    DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1997.SA.V2,DISP=SHR

```

```
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1997.SA.V3,
// DISP=(NEW,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
// DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y98 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//* *-----*
//* |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW98JCL)|
//* *-----*
//*
//* JCL FOR YEAR 1998 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
```

```
//JS098 EXEC SAS9,
// WORK='100000,50000',
// PARM='SYSPARM="1998"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1998.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1998.SA.V3,
// DISP=(NEW,CATLG,DELETE),
// UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
// DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y99 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
// MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//* *-----*
//* |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW99JCL)|
//* *-----*
//*
//* JCL FOR YEAR 1999 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
```

```
//JS099 EXEC SAS9,
// WORK='100000,50000',
// PARM='SYSPARM="1999"'
```

```

// *
// OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y1999.SA.V2,DISP=SHR
// LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
// OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y1999.SA.V3,
//     DISP=(NEW,CATLG,DELETE),
//     UNIT=TSILO
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//     DISP=(SHR,PASS,KEEP)
// *

```

```

// #3836Y00 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//     MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
// *****

```

```

// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW00JCL) |
// *          *-----*

```

```

// * JCL FOR YEAR 2000 TO RUN LDWPATCH
// *
// * CREATED BY: PATSY LLOYD 7/13/12
// * MPR PHONE 202 484-4240
// * E-MAIL PLLOYD@MATHEMATICA-MPR.COM

```

```

// *****

```

```

// JS000 EXEC SAS9,
//     WORK='100000,50000',
//     PARM='SYSPARM="2000"'

```

```

// OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2000.SA.V2,DISP=SHR
// LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
// OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2000.SA.V3,
//     DISP=(NEW,CATLG,DELETE),
//     UNIT=TSILO
// SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//     DISP=(SHR,PASS,KEEP)
// *

```

```

// #3836Y01 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//     MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
// *****

```

```

// *
// *          *-----*
// *          | OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW01JCL) |
// *          *-----*

```

```

// * JCL FOR YEAR 2001 TO RUN LDWPATCH
// *
// * CREATED BY: PATSY LLOYD 7/13/12
// * MPR PHONE 202 484-4240
// * E-MAIL PLLOYD@MATHEMATICA-MPR.COM

```

```

// *****

```

```

// JS001 EXEC SAS9,

```

```
//          WORK='100000,50000',
//          PARM='SYSPARM="2001"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2001.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2001.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y02 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
```

```
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW02JCL)|
//*          *-----*
//*
```

```
//* JCL FOR YEAR 2002 TO RUN LDWPATCH
```

```
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
```

```
//*****
//*
```

```
//JS002 EXEC SAS9,
//          WORK='100000,50000',
//          PARM='SYSPARM="2002"'
//*
```

```
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2002.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2002.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y03 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
```

```
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW03JCL)|
//*          *-----*
//*
```

```
//* JCL FOR YEAR 2003 TO RUN LDWPATCH
```

```
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
```

```
//*****
```

```
//*
//JS003 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2003"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2003.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2003.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y04 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*      *-----*
//*      |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW04JCL)|
//*      *-----*
//*
//* JCL FOR YEAR 2004 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
```

```
//*****
//*
//JS004 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2004"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2004.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2004.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y05 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*      *-----*
//*      |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW05JCL)|
//*      *-----*
//*
//* JCL FOR YEAR 2005 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
```

```

//*
//*****
//*
//JS005 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2005"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2005.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2005.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*

//#3836Y06 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*-----*
//* |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW06JCL)|
//* *-----*
//*
//* JCL FOR YEAR 2006 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//*
//JS006 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2006"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2006.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2006.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*

//#3836Y07 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*-----*
//* |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW07JCL)|
//* *-----*
//*
//* JCL FOR YEAR 2007 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12

```

```
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//*
//JS007 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2007"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2007.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2007.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y08 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*      *-----*
//*      |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW08JCL)|
//*      *-----*
//*
//* JCL FOR YEAR 2008 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
```

```
//*****
//*
//JS008 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2008"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2008.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2008.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*
```

```
//#3836Y09 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//      MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*      *-----*
//*      |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW09JCL)|
//*      *-----*
//*
//* JCL FOR YEAR 2009 TO RUN LDWPATCH
```

```

//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//*
//JS009 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2009"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2009.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2009.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//*

//#3836Y10 JOB (12510000,T715,,SAS,,ITC9FL),PATSY-LLOYD,
//          MSGCLASS=1,MSGLEVEL=(1,1),NOTIFY=#3836
//*****
//*
//*          *-----*
//*          |OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDW10JCL)|
//*          *-----*
//*
//* JCL FOR YEAR 2010 TO RUN LDWPATCH
//*
//* CREATED BY: PATSY LLOYD 7/13/12
//* MPR PHONE 202 484-4240
//* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
//*
//*****
//JS010 EXEC SAS9,
//      WORK='100000,50000',
//      PARM='SYSPARM="2010"'
//*
//OLD DD DSN=OPDR.TG.PRD.ETTW.#2127.TRF10P.Y2010.SA.V2,DISP=SHR
//LDW DD DSN=OPDR.TG.PRD.ETTW.N4671.TRF10.LDWSTRNG.SA.V2,DISP=SHR
//OUT DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10P.Y2010.SA.V3,
//      DISP=(NEW,CATLG,DELETE),
//      UNIT=TSILO
//SYSIN DD DSN=OPDR.TG.PRD.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH),
//      DISP=(SHR,PASS,KEEP)
//

```

Appendix A.93
SAS Code: LDWPATCH

```
*****
*
*          *-----*
*          | OPDR.TG.PR.D.ETTW.#3836.TRF10.ANN.PRDLIB(LDWPATCH) |
*          *-----*
*
* MERGE NEW LDW VARIABLES ONTO THE ANNUAL FILES
* CREATED BY: PATSY LLOYD 7/13/2012
* MPR PHONE 202 484-4240
* E-MAIL PLLOYD@MATHEMATICA-MPR.COM
*****;

OPTIONS NOCENTER LS=132 PS=60 COMPRESS=BINARY   OBS=MAX;

***TWO DIGIT YEAR MACRO VARIABLE;
%LET YR = SUBSTR(&SYSPARM,3,2);

*** CHANGE ENDYR AND ENDMN FOR EACH RUN;
%let begyr=&SYSPARM;
%let endyr=&SYSPARM; /* change as needed */
%let endmn=12;      /* change as needed */

***ASSIGN MACRO VARIABLES TO HANDLE TIME SERIES DATA;
%macro start;
%let k=1;
%do i=&begyr %to &endyr;
  %if &i<2000 %then %let yr=%eval(&i-1900);
  %else %IF %eval(&i-2000)<10 %THEN %let yr=0%eval(&i-2000);
  %else %let yr=%eval(&i-2000);
  %if &i=&endyr %then %let emn=%eval(&endmn);
  %else %let emn=12;
  %do j=1 %to &emn;
    %if &j<10 %then %let mn=0%eval(&j);
    %else %let mn=%eval(&j);
    %let x&k=%eval(&yr)%eval(&mn);
    %let k=%eval(&k+1);
  %end;
%end;
%let tot=%eval(&k-1);

DATA OUT.Y&SYSPARM;
  MERGE OLD.Y&SYSPARM (DROP = LDWSSI: LDWDI: LDWCM:)
        LDW.LDWSTRNG (KEEP = LDWSSI&YR.:
                        LDWDI&YR.:
                        LDWCM&YR.:
                        SSN);
  BY SSN;

LABEL
%DO I=1 %TO &TOT;
  LDWSSI&&X&I = "&&X&I SSI LEFT DUE TO WORK INDICATOR"
  LDWDI&&X&I  = "&&X&I SSDI LEFT DUE TO WORK INDICATOR"
  LDWCM&&X&I  = "&&X&I COMBINED LEFT DUE TO WORK INDICATOR"
```

```
%END;  
;
```

```
RUN;
```

```
%MEND;  
%START;
```

```
PROC CONTENTS DATA=OUT.Y&SYSPARM;  
TITLE "CONTENTS OF YEAR Y&SYSPARM";  
RUN;
```

```
PROC FREQ DATA=OUT.Y&SYSPARM;  
  TABLES LDWDI: LDWSSI: LDWCM: /LIST MISSING;  
RUN;
```

APPENDIX B
TABLES/CHARTS

Table 1. Age Distribution of DI Beneficiaries in SSA Compared with Results from TRF10, All Disabled Workers

	Under 30	30-34	35-39	40-44	45-49	50-54	55-59	60-FRA
TRF10	2.9%	3.5%	4.9%	7.7%	12.6%	18.3%	23.1%	26.9%
SSA	2.9%	3.4%	4.8%	7.5%	12.0%	17.5%	22.2%	29.6%

Source: SSA Statistics based on published information from SSA DI Table 19 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Notes: FRA = full retirement age.

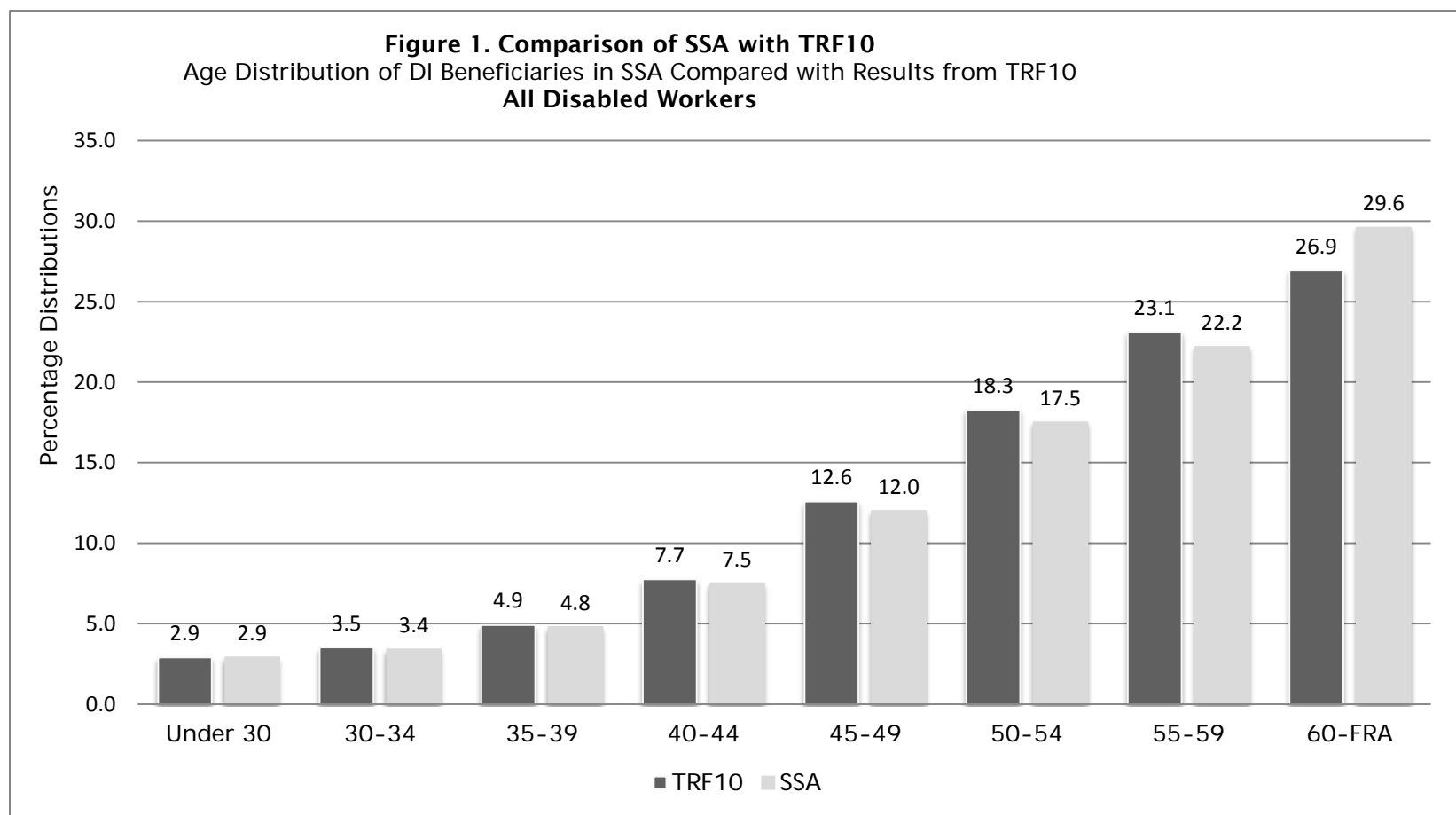


Table 2. Age Distribution of DI Beneficiaries in SSA Compared with Results from TRF10, Male Disabled Workers

	Under 30	30-34	35-39	40-44	45-49	50-54	55-59	60-FRA
TRF10	3.2%	3.5%	4.6%	7.5%	12.2%	18.1%	23.2%	27.9%
SSA	3.1%	3.4%	4.6%	7.3%	11.8%	17.3%	22.1%	30.4%

Source: SSA Statistics based on published information from SSA DI Table 19 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Notes: FRA = full retirement age.

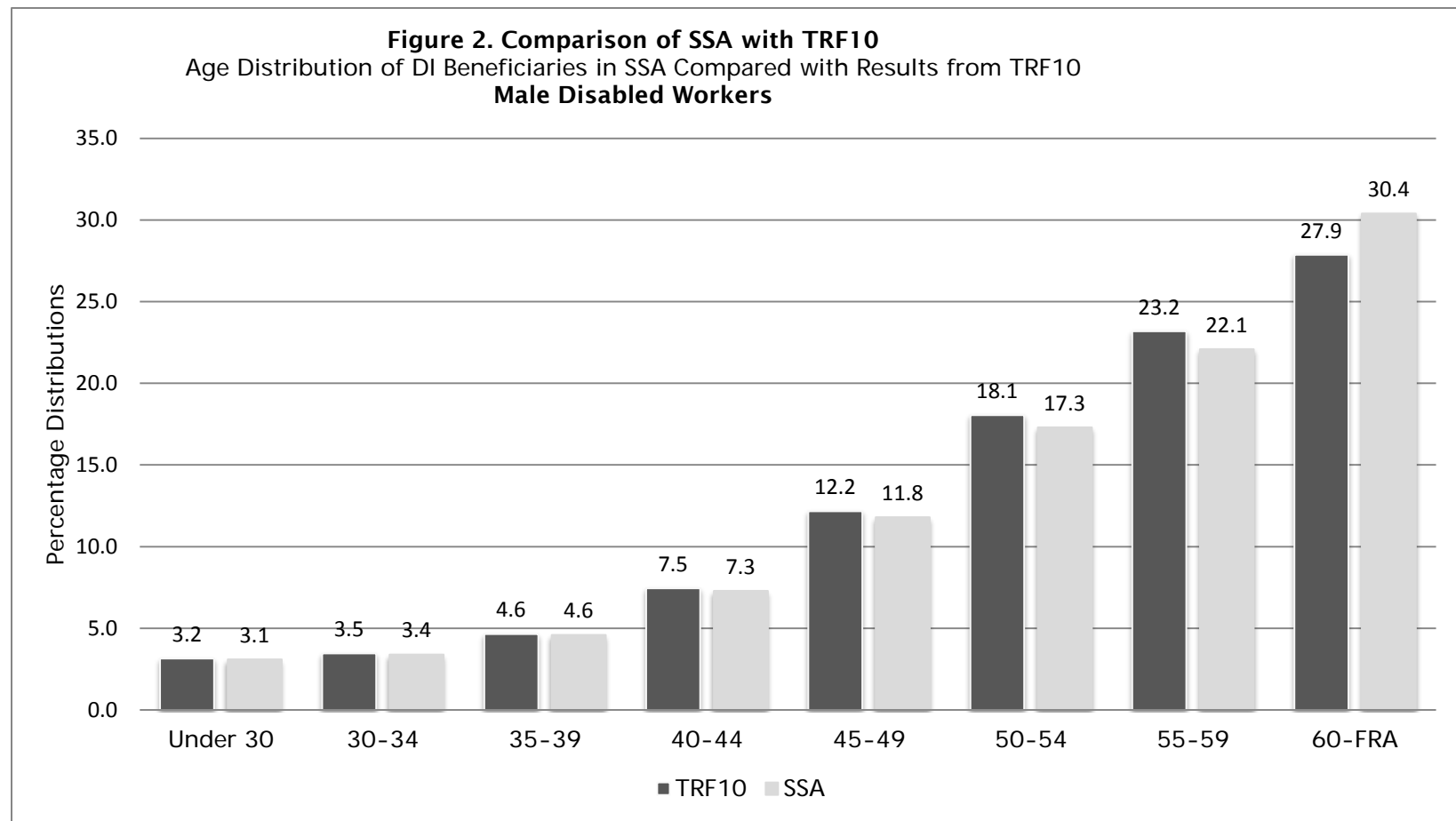


Table 3. Age Distribution of DI Beneficiaries in SSA Compared with Results from TRF10, Female Disabled Workers

	Under 30	30-34	35-39	40-44	45-49	50-54	55-59	60-FRA
TRF10	2.6%	3.6%	5.2%	8.1%	13.0%	18.5%	23.0%	25.9%
SSA	2.6%	3.5%	5.0%	7.7%	12.3%	17.8%	22.3%	28.7%

Source: SSA Statistics based on published information from SSA DI Table 19 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Notes: FRA = full retirement age.

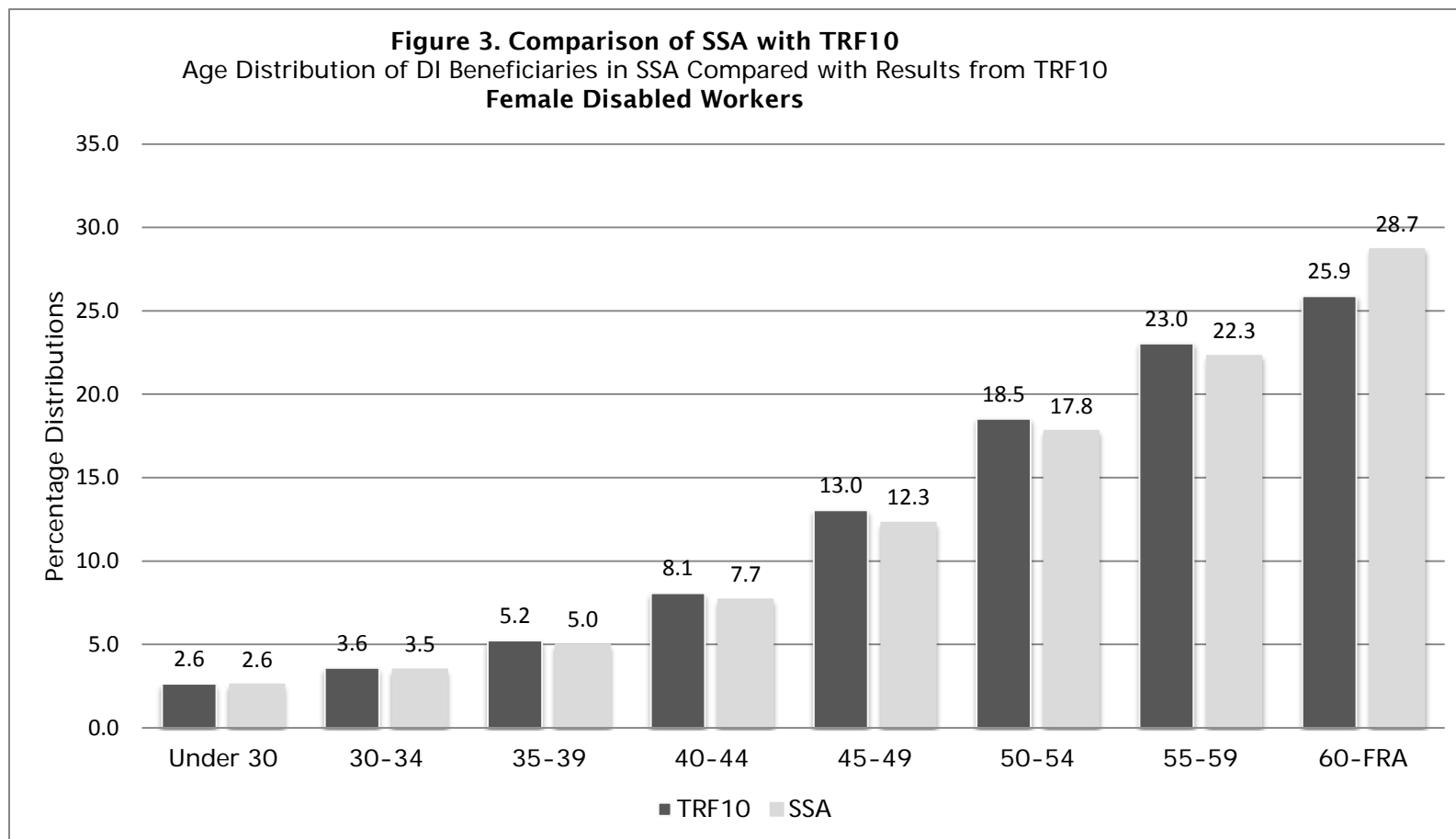


Table 4. Distribution of Diagnostic Groups, DI Beneficiaries	TRF10	SSA
Congenital anomalies	0.2%	0.2%
Endocrine, nutritional, and metabolic diseases	3.5%	3.5%
Infectious and parasitic diseases	1.6%	1.5%
Injuries	4.2%	4.2%
Intellectual disability	4.1%	4.5%
Other mental disorders	28.9%	28.3%
Neoplasms	3.0%	3.1%
Blood and blood-forming organs diseases	0.3%	0.3%
Circulatory system diseases	8.5%	8.6%
Digestive system diseases	1.7%	1.6%
Genitourinary system diseases	1.7%	1.7%
Musculoskeletal system and connective tissue diseases	28.2%	28.2%
Nervous system and sense organs diseases	9.4%	9.4%
Respiratory system diseases	2.8%	2.9%
Skin and subcutaneous tissue diseases	0.2%	0.2%
Other diseases	0.2%	0.2%
Unknown diseases	1.4%	1.5%

Source: SSA statistics based on published information from SSA DI Table 21 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 4. Comparison of SSA with TRF10
 Distribution of Diagnostic Groups, DI Beneficiaries

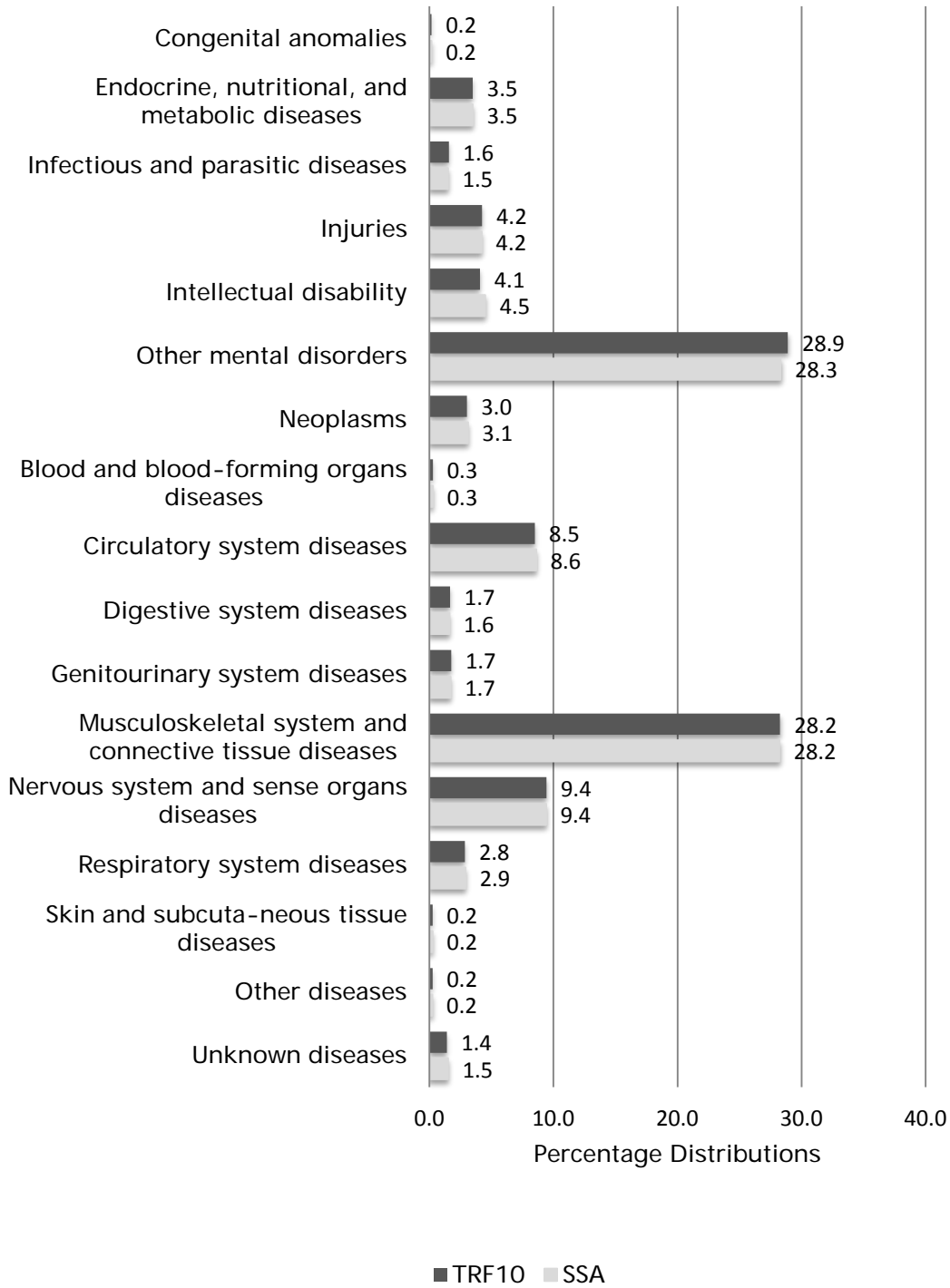


Table 5a: Percentage distribution of all DI Disabled Workers, by state for TRF10 and the SSA

State or area	TRF10	SSA
Alabama	2.64%	2.63%
Alaska	0.15%	0.15%
Arizona	1.75%	1.76%
Arkansas	1.63%	1.63%
California	8.10%	8.07%
Colorado	1.15%	1.15%
Connecticut	0.93%	0.94%
Delaware	0.31%	0.32%
District of Columbia	0.16%	0.16%
Florida	5.92%	5.95%
Georgia	3.01%	3.03%
Hawaii	0.29%	0.28%
Idaho	0.47%	0.47%
Illinois	3.32%	3.35%
Indiana	2.26%	2.28%
Iowa	0.85%	0.87%
Kansas	0.84%	0.84%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 5a. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
All Disabled Workers

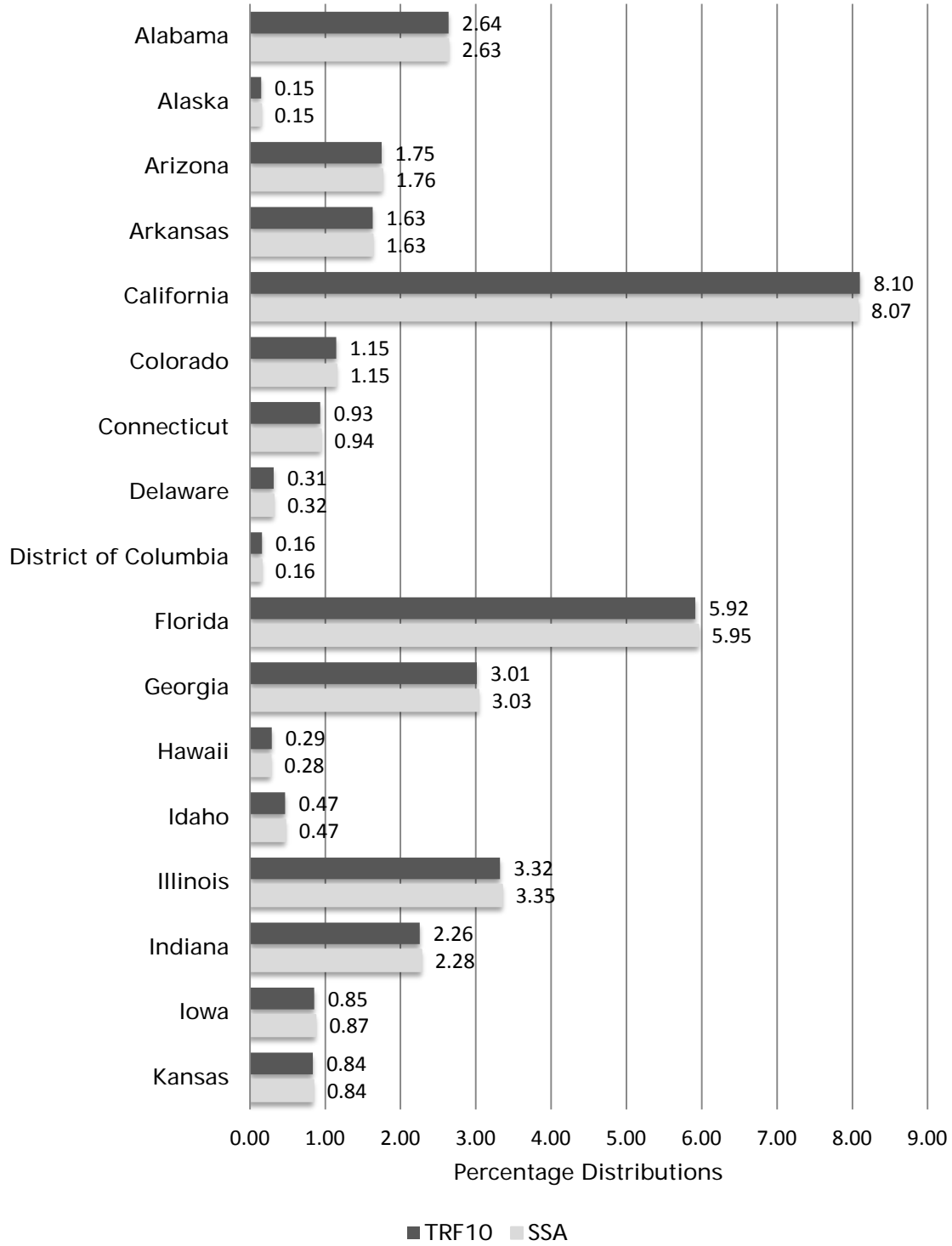


Table 5b: Percentage distribution of all DI Disabled Workers, by state for TRF10 and the SSA

State or area	TRF10	SSA
Kentucky	2.45%	2.43%
Louisiana	1.74%	1.74%
Maine	0.68%	0.68%
Maryland	1.41%	1.41%
Massachusetts	2.30%	2.28%
Michigan	3.83%	3.84%
Minnesota	1.39%	1.41%
Mississippi	1.53%	1.53%
Missouri	2.45%	2.47%
Montana	0.31%	0.32%
Nebraska	0.48%	0.49%
Nevada	0.67%	0.68%
New Hampshire	0.52%	0.52%
New Jersey	2.25%	2.27%
New Mexico	0.72%	0.71%
New York	5.98%	5.98%
North Carolina	3.75%	3.78%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 5b. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
All Disabled Workers

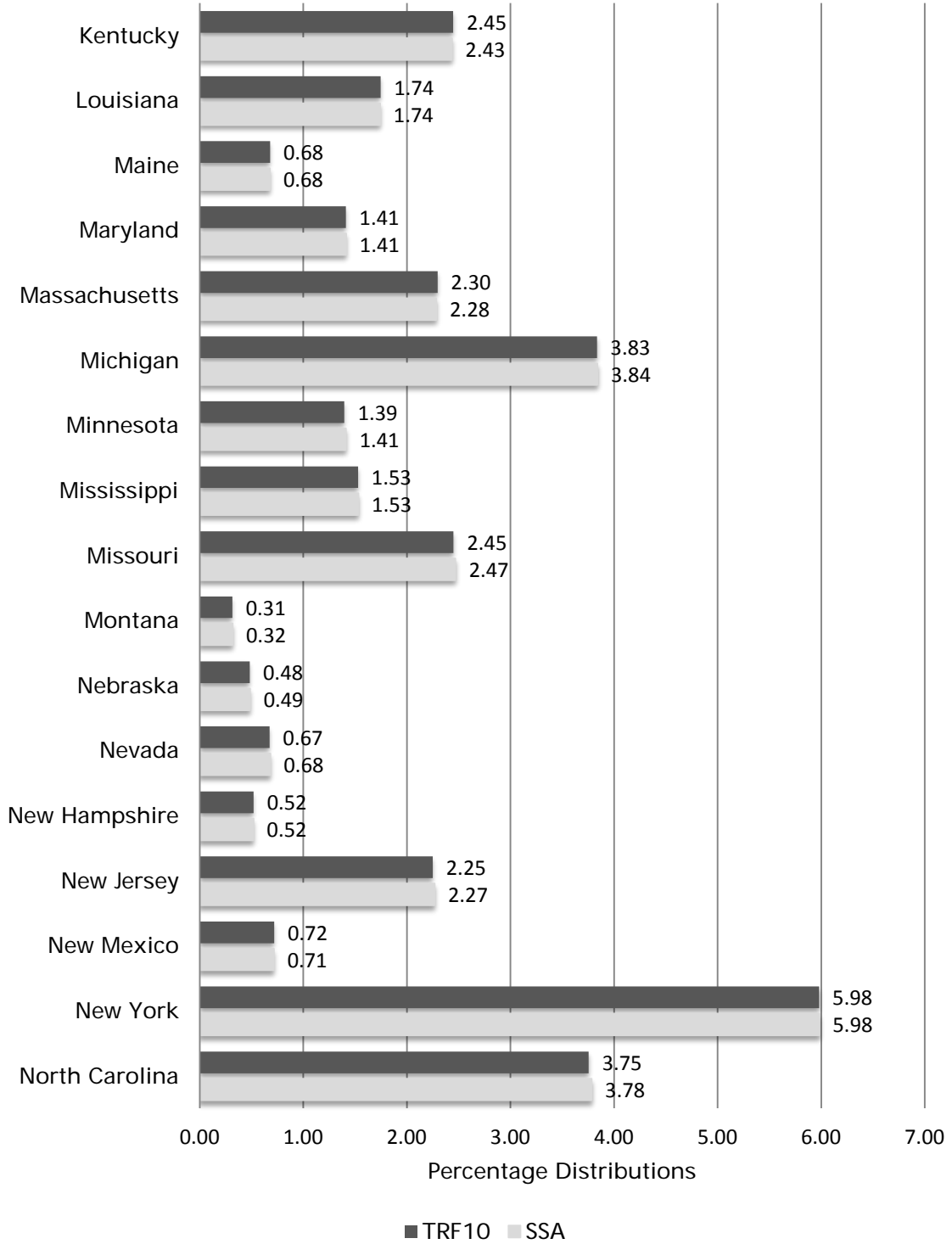


Table 5c: Percentage distribution of all DI Disabled Workers, by state for TRF10 and the SSA

State or area	TRF10	SSA
North Dakota	0.16%	0.17%
Ohio	3.81%	3.82%
Oklahoma	1.46%	1.47%
Oregon	1.18%	1.19%
Pennsylvania	4.54%	4.54%
Rhode Island	0.42%	0.42%
South Carolina	1.99%	2.01%
South Dakota	0.21%	0.22%
Tennessee	2.79%	2.81%
Texas	6.43%	6.44%
Utah	0.51%	0.51%
Vermont	0.25%	0.25%
Virginia	2.43%	2.44%
Washington	1.94%	1.95%
West Virginia	1.14%	1.14%
Wisconsin	1.77%	1.79%
Wyoming	0.14%	0.14%
Territories Total	2.10%	2.12%
Other	0.49%	0.16%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 5c. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
All Disabled Workers

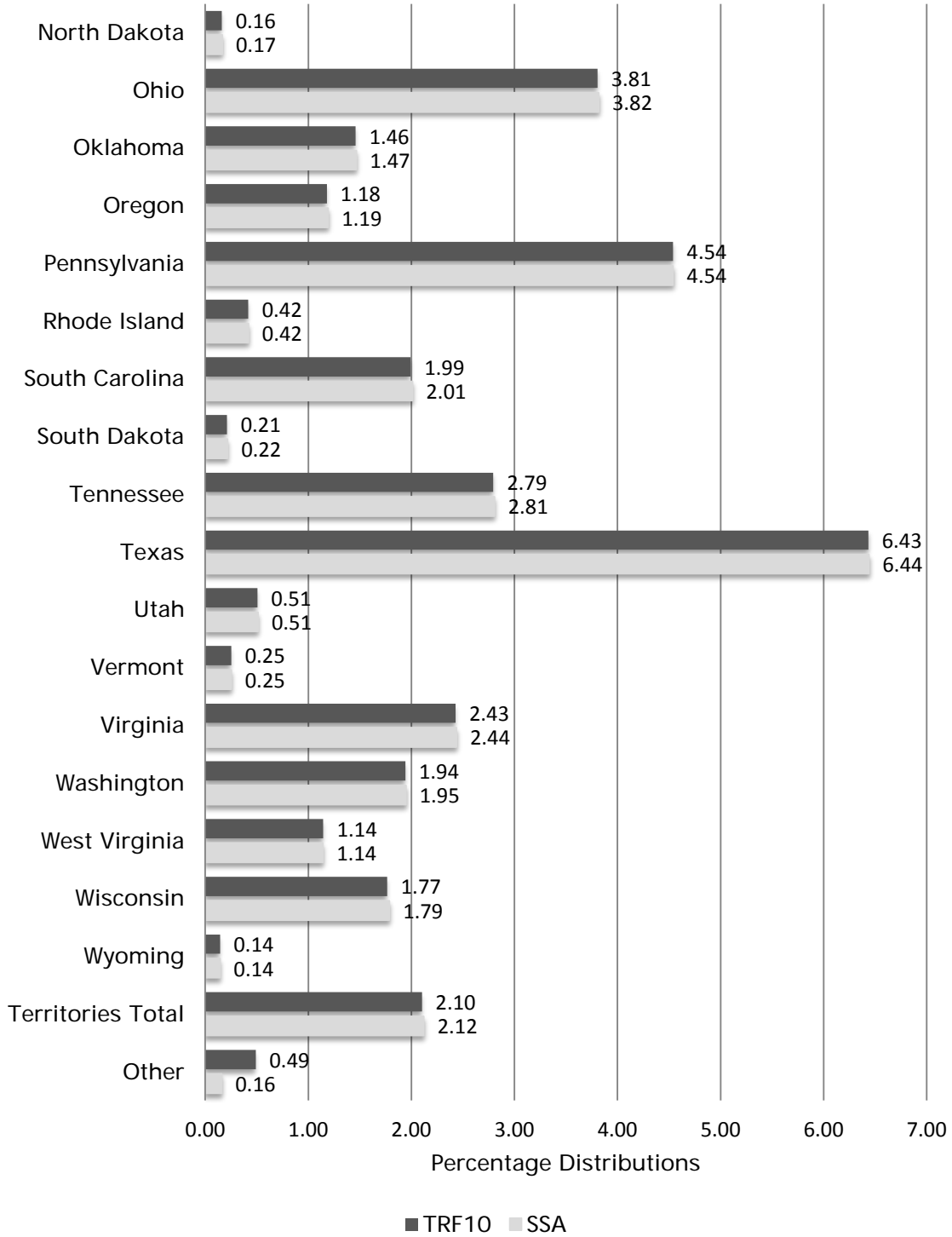


Table 6a: Percentage distribution of Male DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Alabama	2.57%	2.55%
Alaska	0.16%	0.16%
Arizona	1.72%	1.74%
Arkansas	1.63%	1.63%
California	8.24%	8.23%
Colorado	1.14%	1.14%
Connecticut	0.90%	0.92%
Delaware	0.29%	0.29%
District of Columbia	0.16%	0.16%
Florida	5.94%	5.97%
Georgia	2.92%	2.93%
Hawaii	0.32%	0.30%
Idaho	0.48%	0.48%
Illinois	3.28%	3.31%
Indiana	2.21%	2.23%
Iowa	0.86%	0.88%
Kansas	0.80%	0.81%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 6a. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Male Disabled Workers

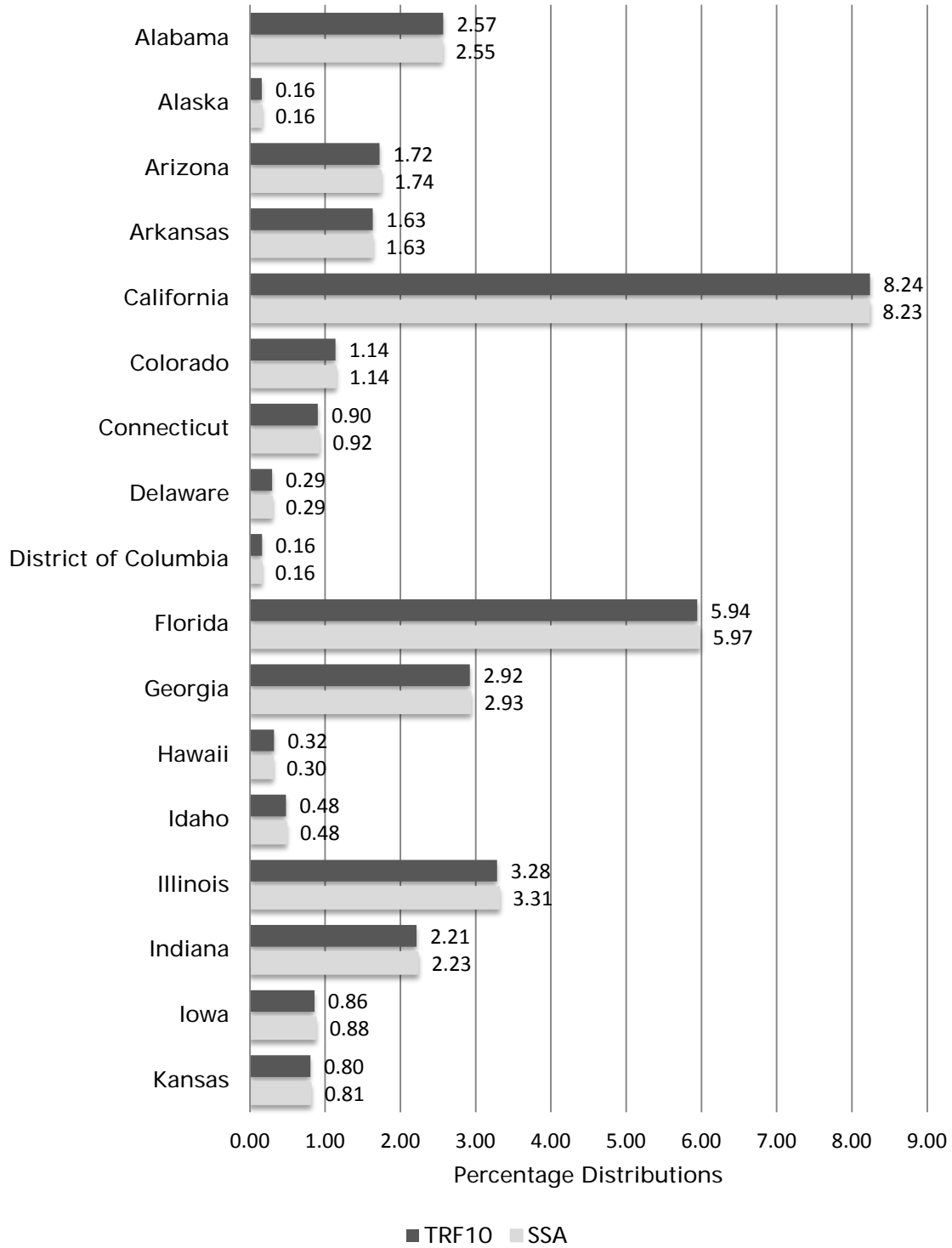


Table 6b: Percentage distribution of Male DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Kentucky	2.60%	2.58%
Louisiana	1.94%	1.94%
Maine	0.71%	0.71%
Maryland	1.35%	1.35%
Massachusetts	2.23%	2.23%
Michigan	3.80%	3.81%
Minnesota	1.38%	1.40%
Mississippi	1.52%	1.51%
Missouri	2.42%	2.44%
Montana	0.33%	0.34%
Nebraska	0.46%	0.46%
Nevada	0.67%	0.67%
New Hampshire	0.48%	0.49%
New Jersey	2.16%	2.19%
New Mexico	0.76%	0.76%
New York	5.96%	5.97%
North Carolina	3.58%	3.60%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 6b. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Male Disabled Workers

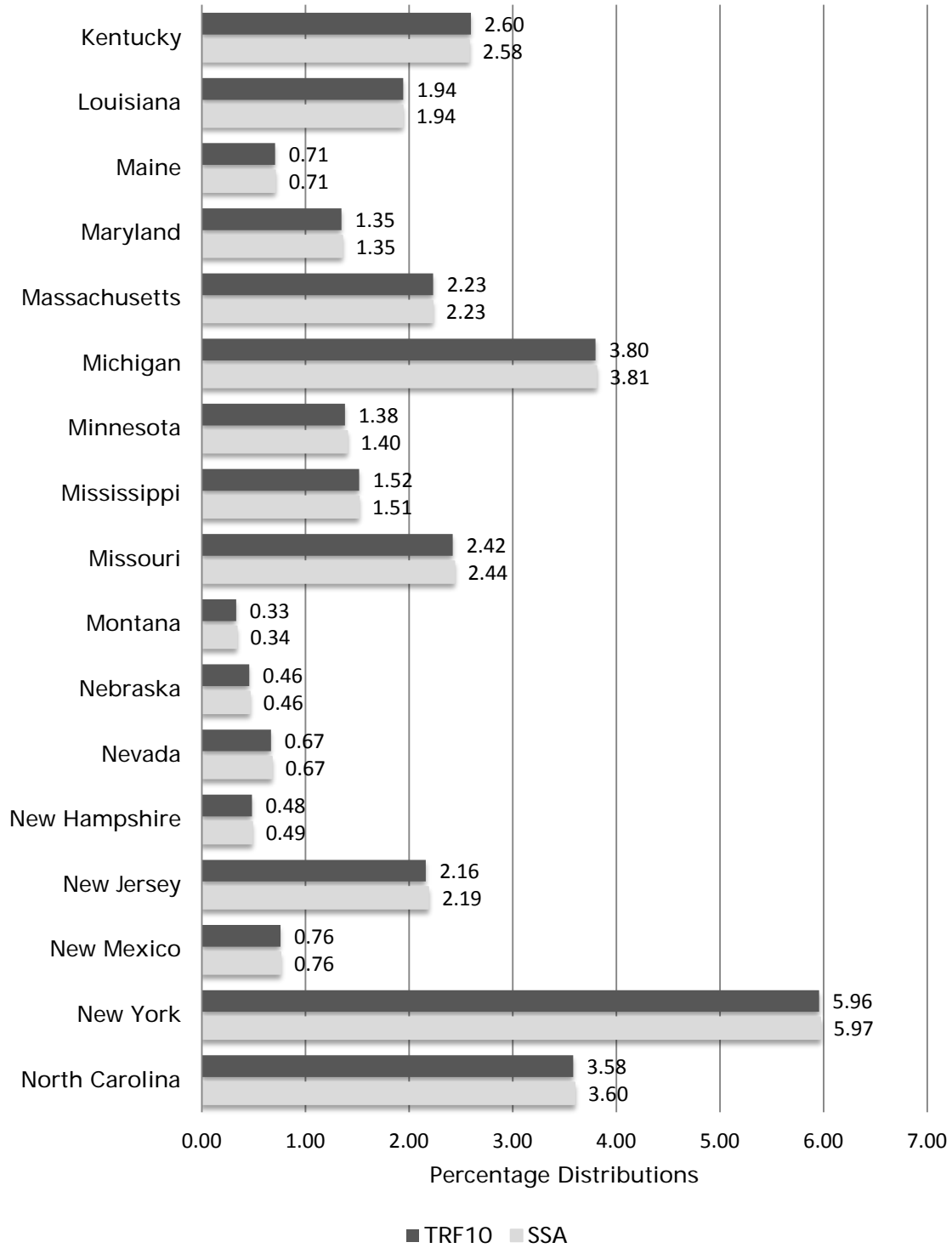


Table 6c: Percentage distribution of Male DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
North Dakota	0.17%	0.17%
Ohio	3.87%	3.88%
Oklahoma	1.45%	1.45%
Oregon	1.21%	1.22%
Pennsylvania	4.52%	4.52%
Rhode Island	0.41%	0.41%
South Carolina	1.92%	1.93%
South Dakota	0.21%	0.22%
Tennessee	2.72%	2.72%
Texas	6.47%	6.48%
Utah	0.49%	0.50%
Vermont	0.25%	0.25%
Virginia	2.38%	2.38%
Washington	1.95%	1.96%
West Virginia	1.34%	1.33%
Wisconsin	1.73%	1.76%
Wyoming	0.15%	0.15%
Territories Total	2.24%	2.27%
Other	0.57%	0.22%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 6c. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Male Disabled Workers

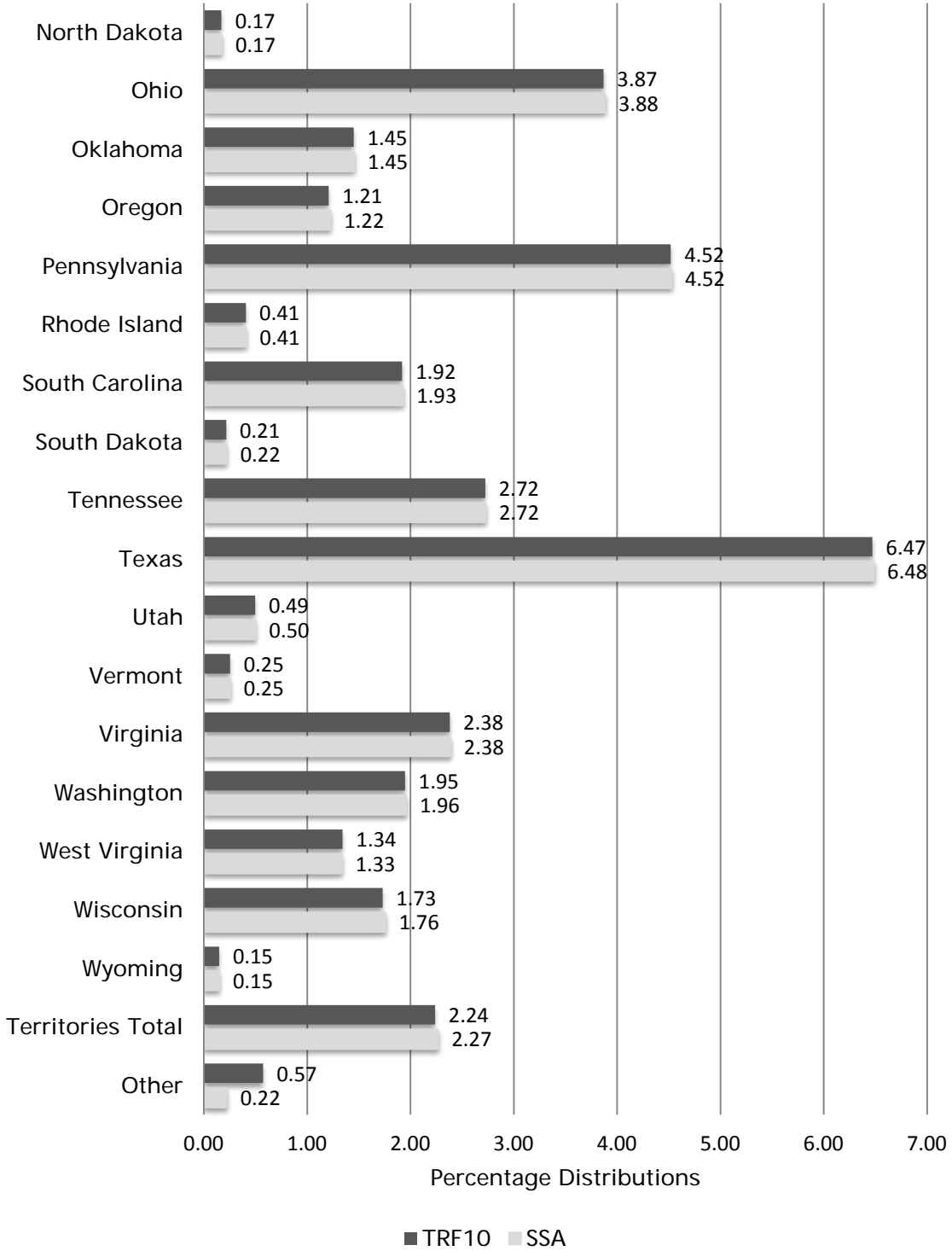


Table 7a: Percentage distribution of Female DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Alabama	2.71%	2.72%
Alaska	0.14%	0.14%
Arizona	1.78%	1.78%
Arkansas	1.62%	1.63%
California	7.95%	7.90%
Colorado	1.16%	1.16%
Connecticut	0.97%	0.97%
Delaware	0.34%	0.34%
District of Columbia	0.16%	0.16%
Florida	5.88%	5.93%
Georgia	3.12%	3.15%
Hawaii	0.26%	0.25%
Idaho	0.46%	0.46%
Illinois	3.37%	3.39%
Indiana	2.30%	2.33%
Iowa	0.85%	0.87%
Kansas	0.87%	0.88%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 7a. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Female Disabled Workers

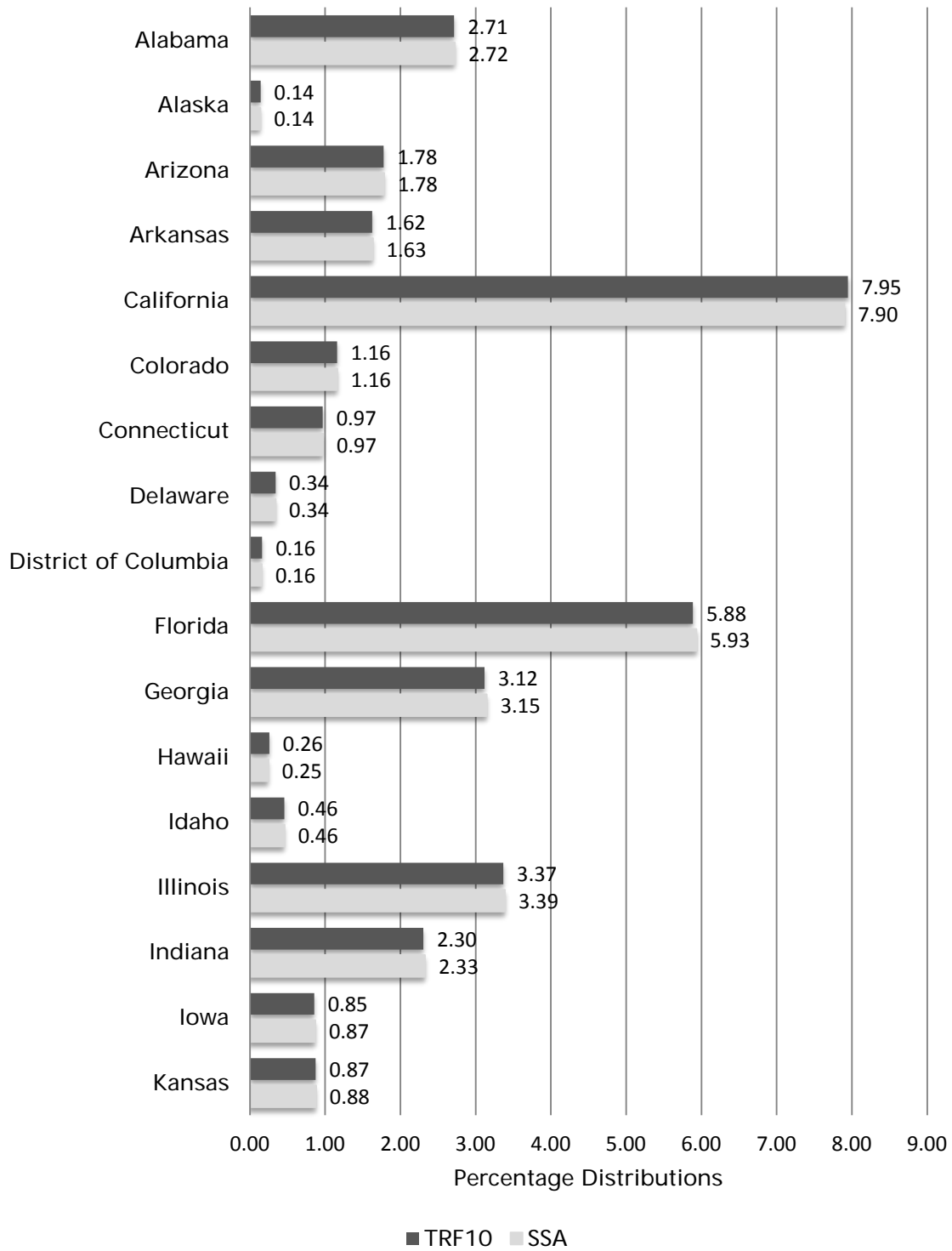


Table 7b: Percentage distribution of Female DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Kentucky	2.28%	2.28%
Louisiana	1.52%	1.52%
Maine	0.65%	0.64%
Maryland	1.48%	1.48%
Massachusetts	2.37%	2.35%
Michigan	3.88%	3.88%
Minnesota	1.41%	1.42%
Mississippi	1.54%	1.55%
Missouri	2.48%	2.50%
Montana	0.29%	0.30%
Nebraska	0.51%	0.51%
Nevada	0.68%	0.69%
New Hampshire	0.56%	0.55%
New Jersey	2.35%	2.36%
New Mexico	0.67%	0.67%
New York	6.00%	6.00%
North Carolina	3.94%	3.99%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 7b. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Female Disabled Workers

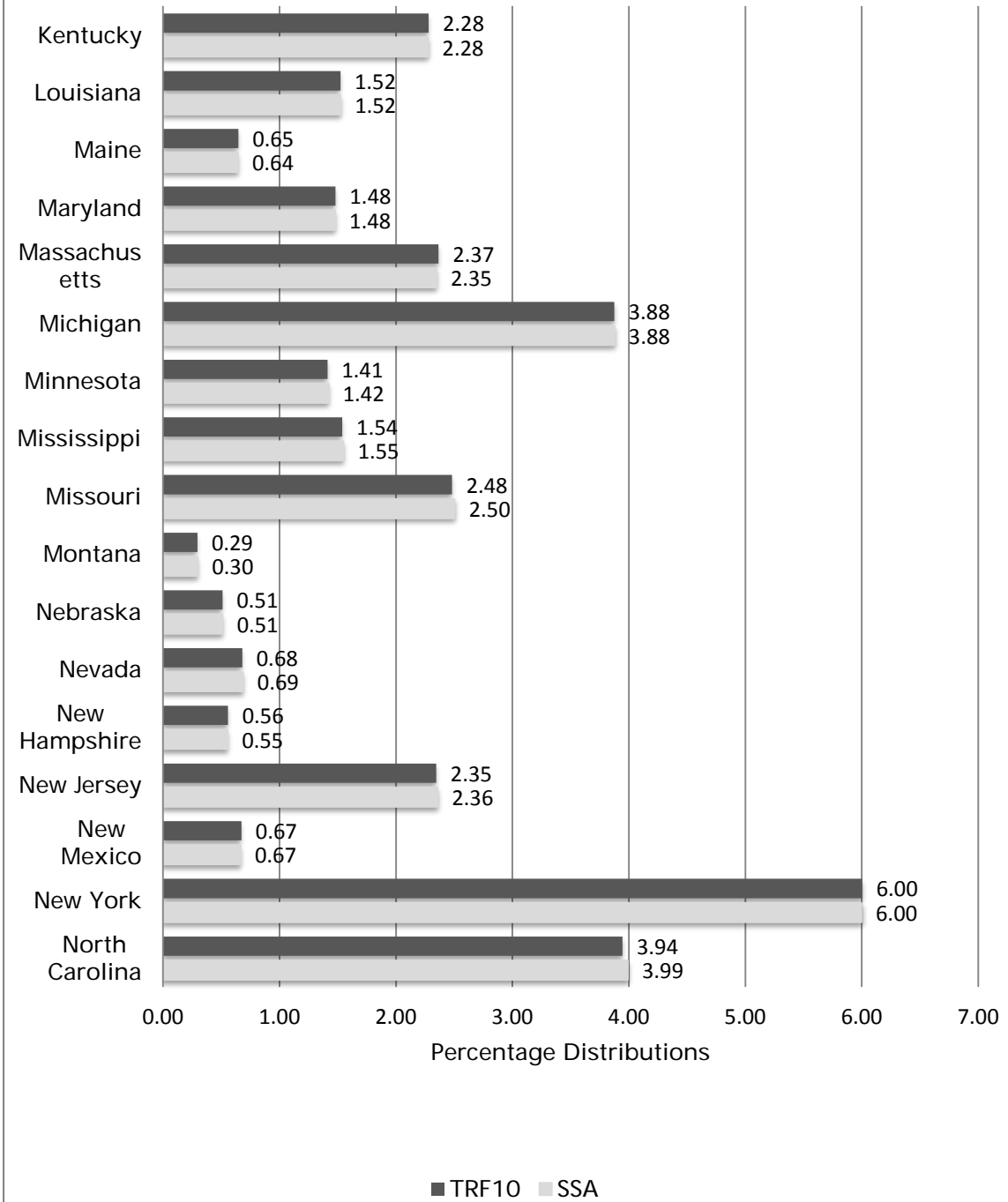


Table 7c: Percentage distribution of Female DI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
North Dakota	0.15%	0.16%
Ohio	3.73%	3.75%
Oklahoma	1.47%	1.48%
Oregon	1.15%	1.16%
Pennsylvania	4.56%	4.55%
Rhode Island	0.43%	0.43%
South Carolina	2.08%	2.10%
South Dakota	0.21%	0.21%
Tennessee	2.88%	2.89%
Texas	6.39%	6.39%
Utah	0.52%	0.52%
Vermont	0.25%	0.25%
Virginia	2.48%	2.49%
Washington	1.93%	1.94%
West Virginia	0.93%	0.93%
Wisconsin	1.81%	1.82%
Wyoming	0.14%	0.14%
Territories Total	1.95%	1.96%
Other	0.40%	0.10%

Source: SSA statistics based on published information from SSA DI Table 27 in the SSA Annual Statistical Supplement (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid data for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 7c. Comparison of SSA with TRF10
 Distribution of States for DI Beneficiaries
Female Disabled Workers

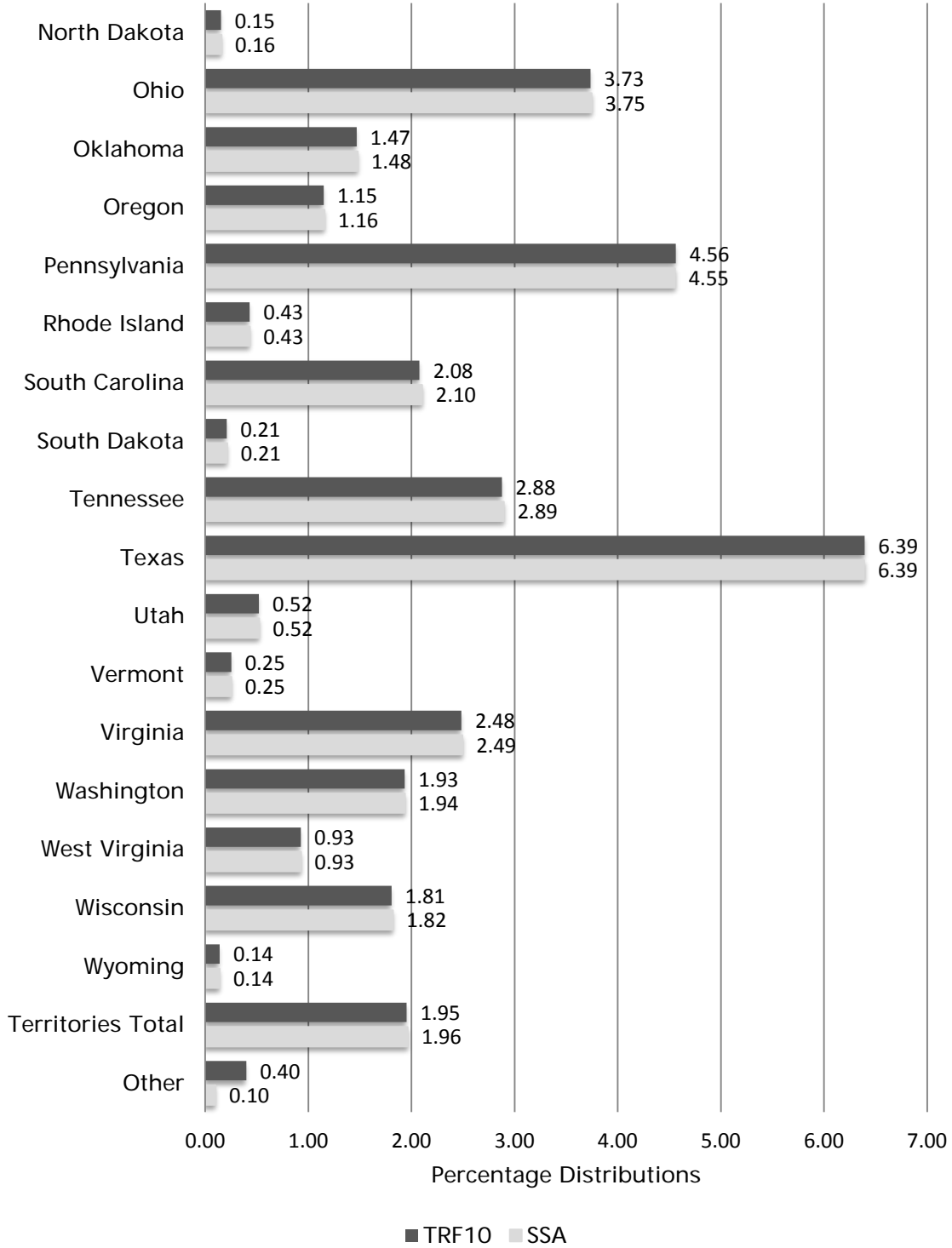


Table 8. Number of Payments to SSI Beneficiaries by Sex, TRF10 and SSA

Number of payments	TRF10	SSA
All payments	4,630,255	4,631,507
Male	2,104,488	2,105,189
Female	2,525,767	2,526,318

Source: SSA statistics based on published information from SSA SSI Table 5 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields of sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

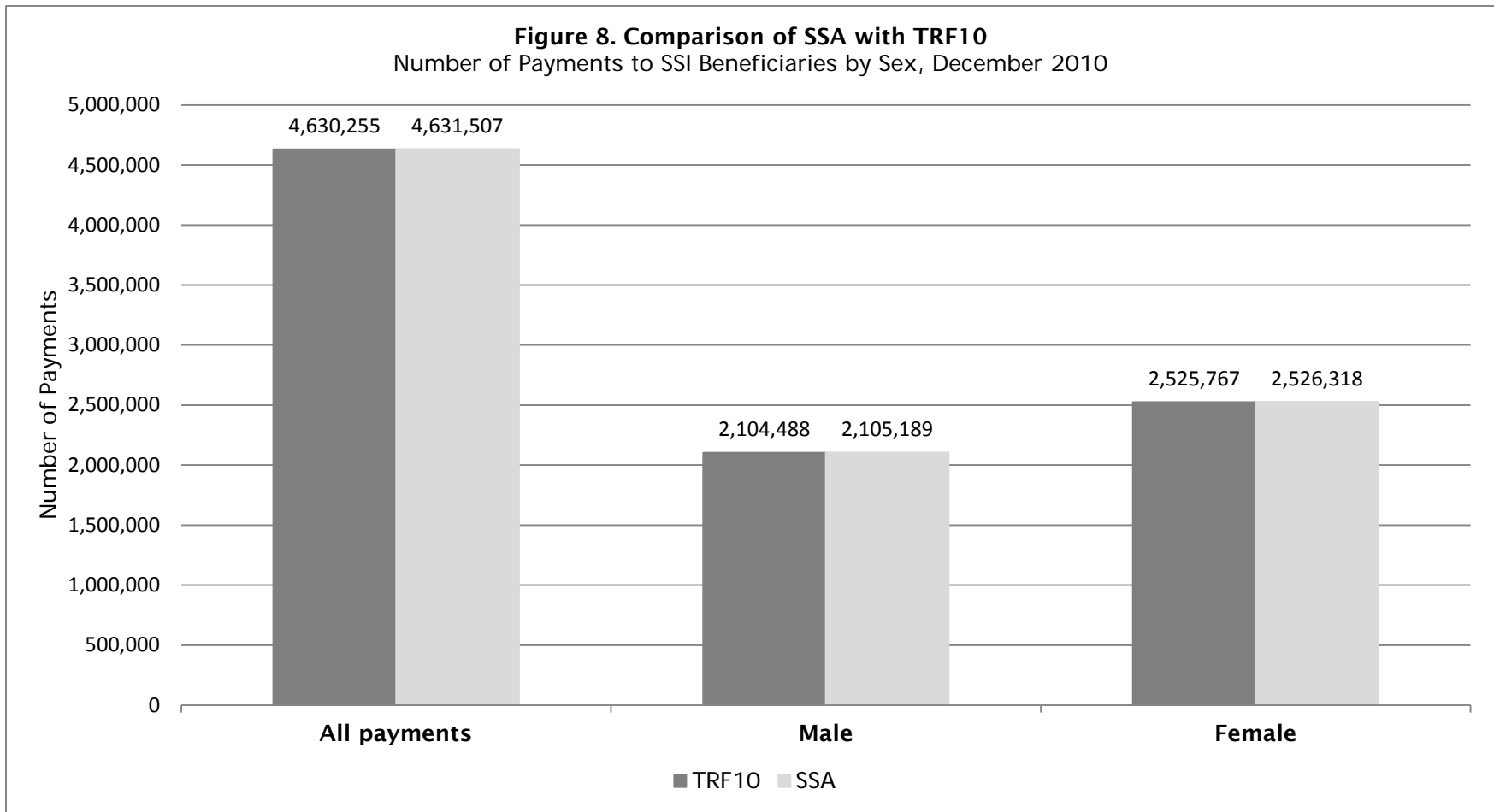


Table 9. Total Payments to SSI Beneficiaries by Sex, TRF10 and SSA

Total payments (thousands of dollars)	TRF10	SSA
All payments	2,553,968	2,663,101
Male	1,168,351	1,228,412
Female	1,385,616	1,434,689

Source: SSA statistics based on published information from SSA SSI Table 5 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields of sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

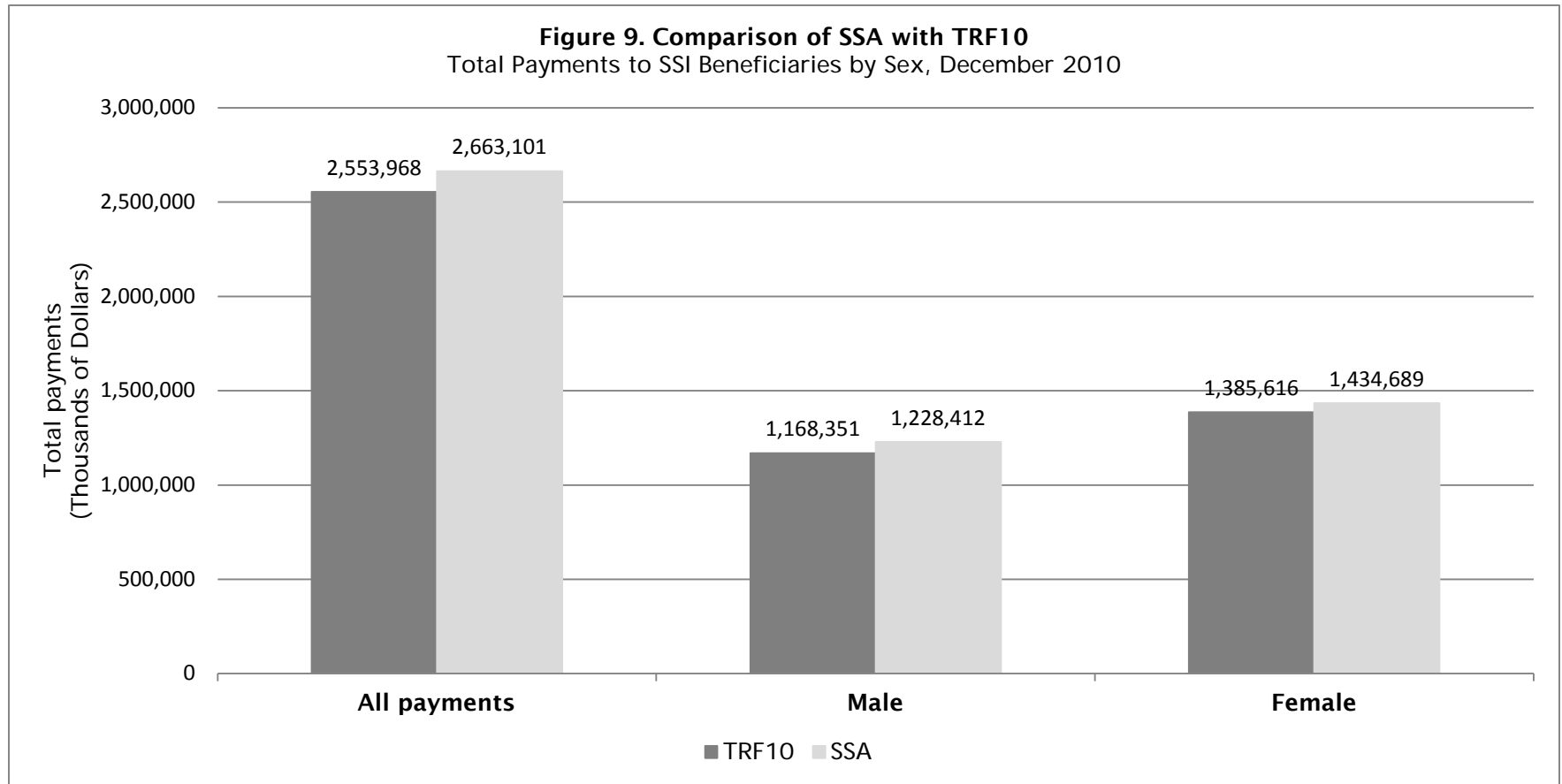


Table 10a: Percentage distribution of all SSI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Alabama	2.42%	2.42%
Alaska	0.17%	0.17%
Arizona	1.36%	1.37%
Arkansas	1.37%	1.38%
California	13.46%	13.24%
Colorado	0.88%	0.90%
Connecticut	0.79%	0.80%
Delaware	0.20%	0.21%
District of Columbia	0.34%	0.34%
Florida	4.98%	5.04%
Georgia	2.94%	2.97%
Hawaii	0.32%	0.32%
Idaho	0.39%	0.39%
Illinois	3.64%	3.64%
Indiana	1.70%	1.72%
Iowa	0.70%	0.71%
Kansas	0.64%	0.66%

Source: SSA statistics based on published information from SSA SSI Table 10 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 10a. Comparison of SSA with TRF10
 Distribution of States for SSI Beneficiaries
All Disabled Workers

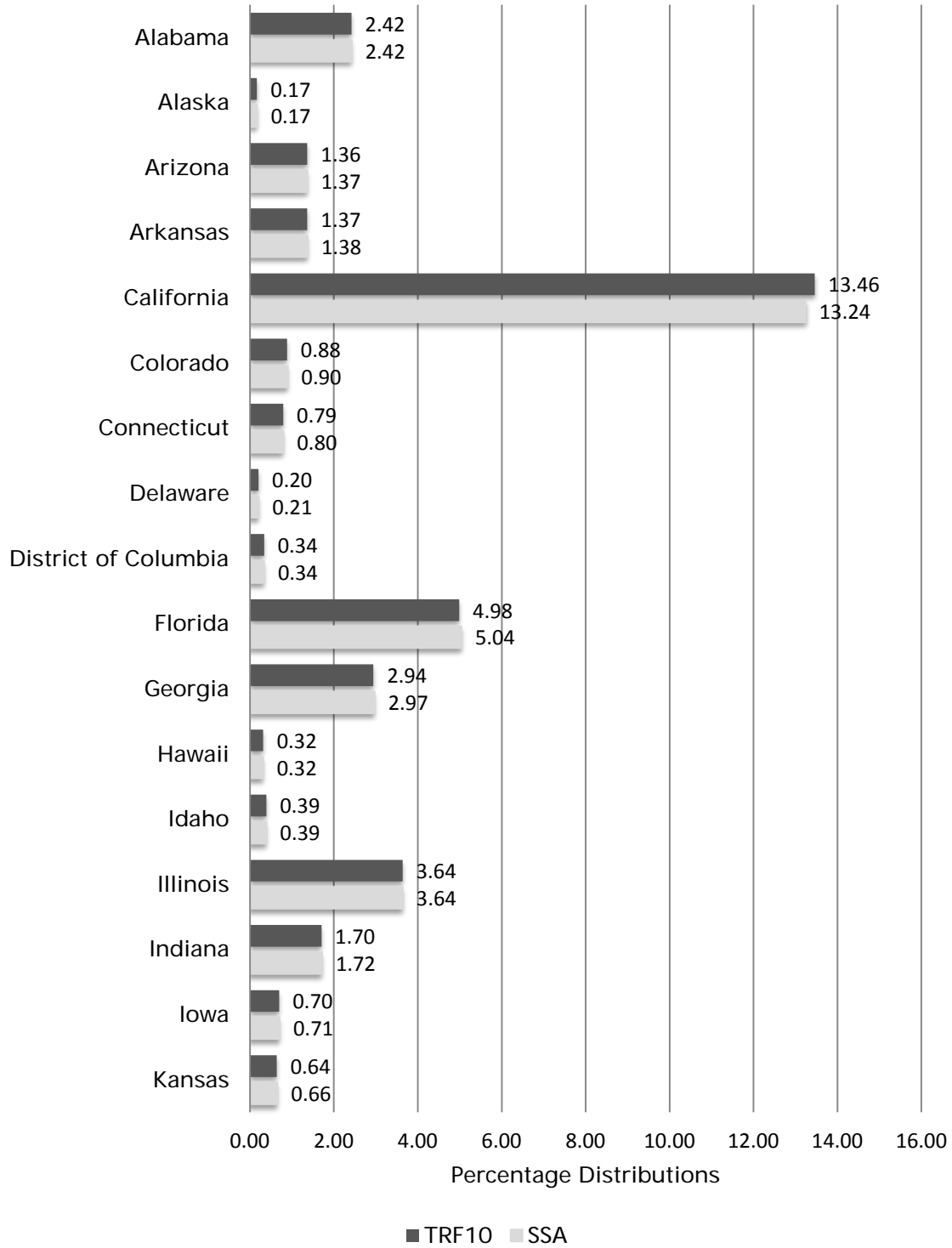


Table 10b: Percentage distribution of all SSI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
Kentucky	2.80%	2.80%
Louisiana	2.31%	2.31%
Maine	0.57%	0.57%
Maryland	1.40%	1.42%
Massachusetts	2.62%	2.57%
Michigan	3.74%	3.75%
Minnesota	1.17%	1.18%
Mississippi	1.63%	1.64%
Missouri	1.95%	1.99%
Montana	0.26%	0.26%
Nebraska	0.37%	0.38%
Nevada	0.48%	0.49%
New Hampshire	0.28%	0.29%
New Jersey	1.98%	1.94%
New Mexico	0.76%	0.76%
New York	8.01%	7.83%
North Carolina	2.86%	2.88%

Source: SSA statistics based on published information from SSA SSI Table 10 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

10b. Comparison of SSA with TRF10
Distribution of States for SSI Beneficiaries
All Disabled Workers

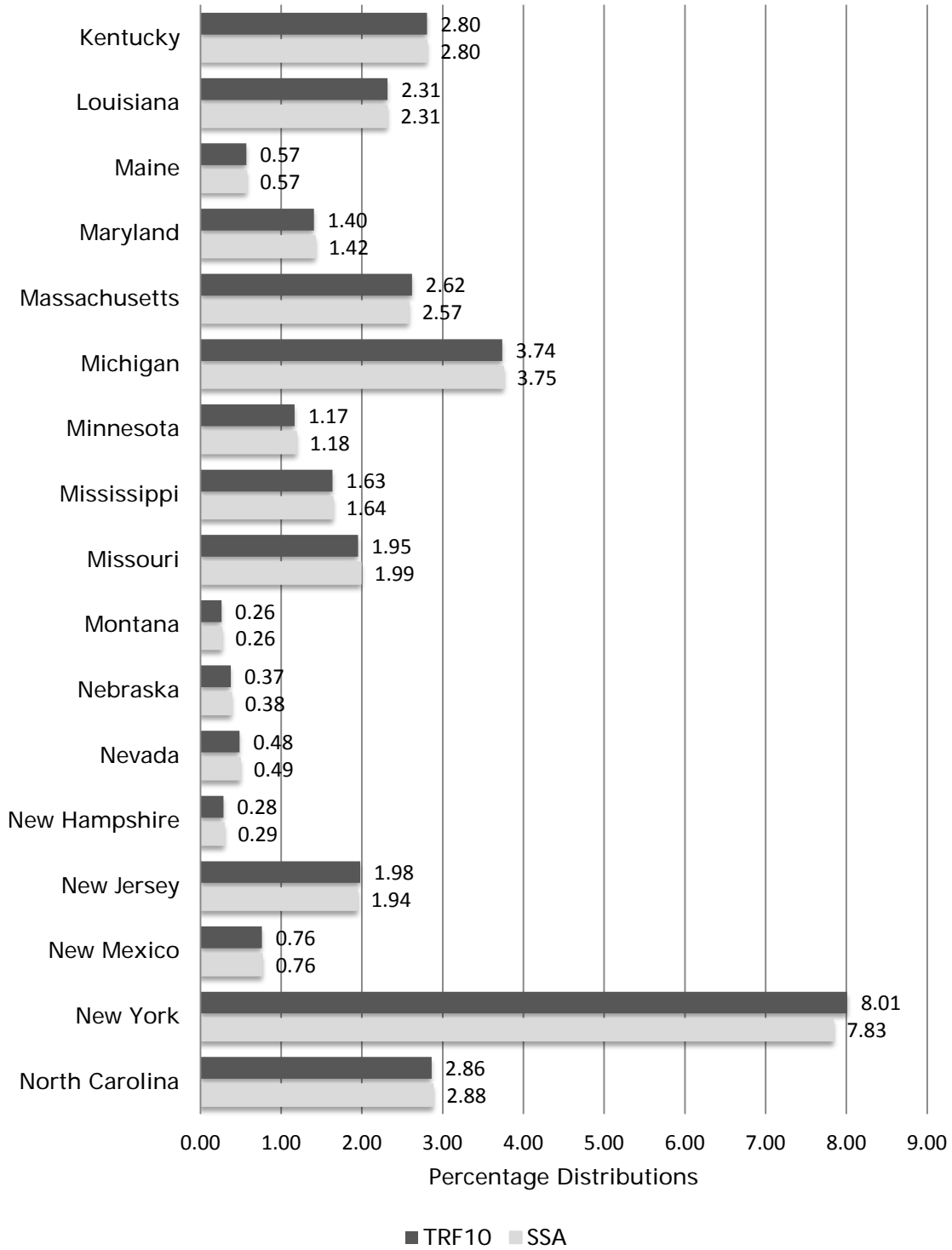


Table 10c: Percentage distribution of all SSI Disabled Workers, by State for TRF10 and the SSA

State or area	TRF10	SSA
North Dakota	0.12%	0.12%
Ohio	4.25%	4.28%
Oklahoma	1.32%	1.33%
Oregon	1.06%	1.07%
Pennsylvania	4.88%	4.89%
Rhode Island	0.46%	0.45%
South Carolina	1.49%	1.50%
South Dakota	0.18%	0.18%
Tennessee	2.51%	2.53%
Texas	6.84%	6.89%
Utah	0.39%	0.39%
Vermont	0.24%	0.24%
Virginia	1.93%	1.95%
Washington	1.92%	1.94%
West Virginia	1.30%	1.29%
Wisconsin	1.52%	1.53%
Wyoming	0.10%	0.10%
Territories Total	0.00%	0.01%
Other	0.00%	0.00%

Source: SSA statistics based on published information from SSA SSI Table 10 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

10c. Comparison of SSA with TRF10
 Distribution of States for SSI Beneficiaries
All Disabled Workers

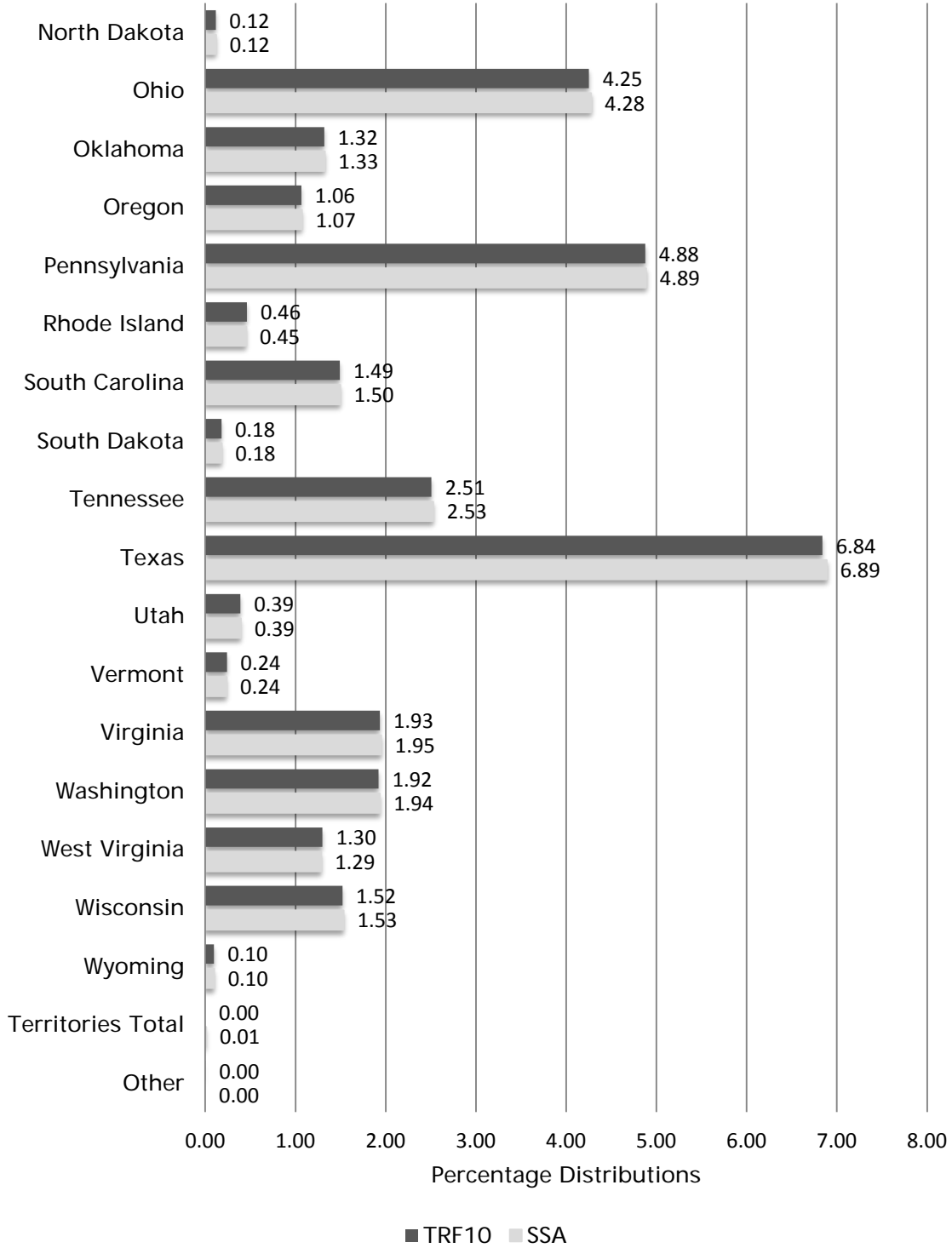


Table 11. All SSI Beneficiaries, by diagnostic group

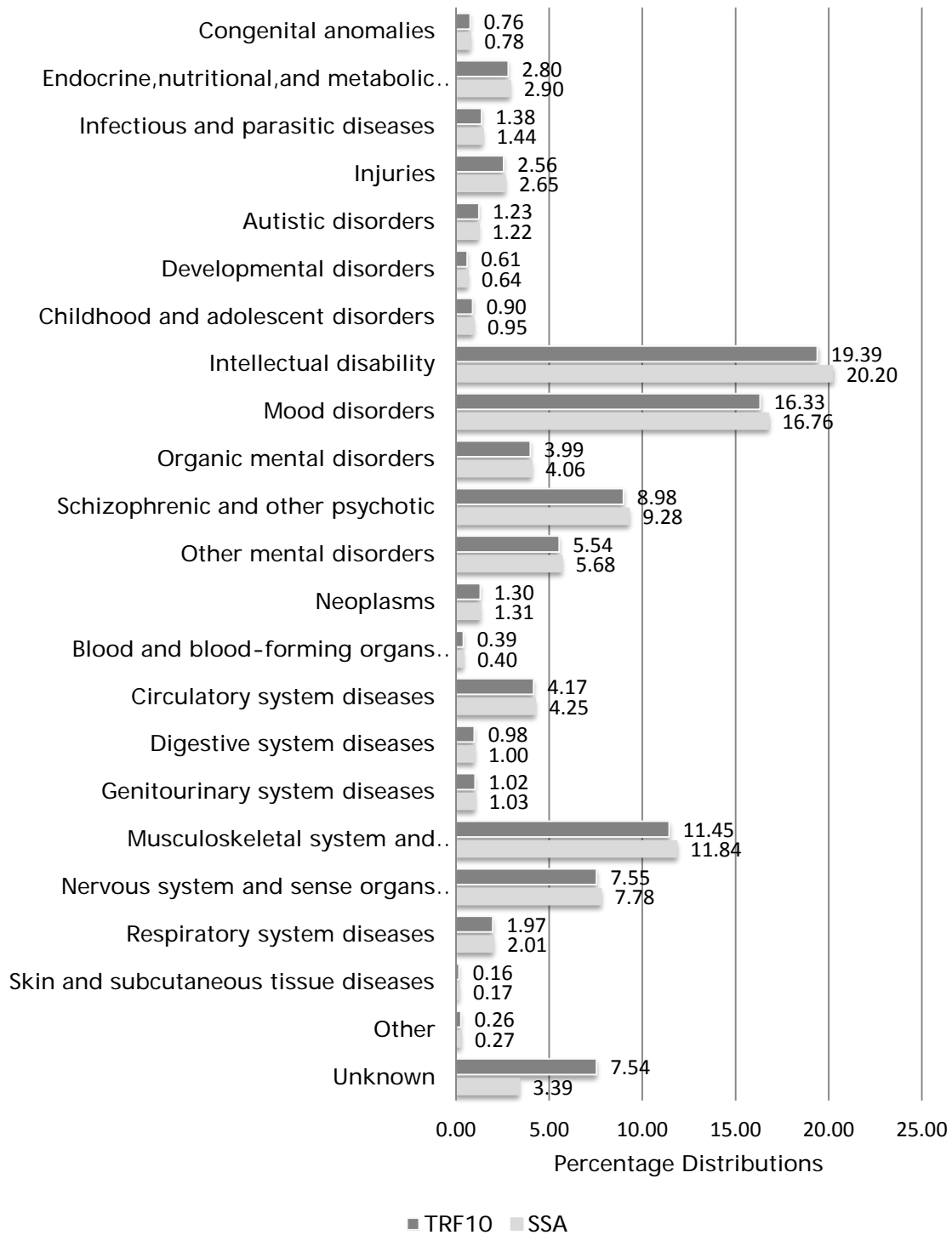
	TRF10	SSA
Congenital anomalies	0.76%	0.78%
Endocrine, nutritional, and metabolic diseases	2.80%	2.90%
Infectious and parasitic diseases	1.38%	1.44%
Injuries	2.56%	2.65%
Autistic disorders	1.23%	1.22%
Developmental disorders	0.61%	0.64%
Childhood and adolescent disorders	0.90%	0.95%
Intellectual disability	19.39%	20.20%
Mood disorders	16.33%	16.76%
Organic mental disorders	3.99%	4.06%
Schizophrenic and other psychotic	8.98%	9.28%
Other mental disorders	5.54%	5.68%
Neoplasms	1.30%	1.31%
Blood and blood-forming organs diseases	0.39%	0.40%
Circulatory system diseases	4.17%	4.25%
Digestive system diseases	0.98%	1.00%
Genitourinary system diseases	1.02%	1.03%
Musculoskeletal system and connective tissue diseases	11.45%	11.84%
Nervous system and sense organs diseases	7.55%	7.78%
Respiratory system diseases	1.97%	2.01%
Skin and subcutaneous tissue diseases	0.16%	0.17%
Other	0.26%	0.27%
Unknown	7.54%	3.39%

Source: SSA statistics based on published information from SSA SSI Table 36 in the SSA Annual Statistical Report (2010). The TRF data are based on TRF10. The TRF data includes beneficiaries with valid fields for sex, beneficiary eligibility code, date of entitlement and a report of current pay status as of December 2010.

Figure 11. Comparison of SSA and TRF10

All SSI Beneficiaries, by diagnostic group

All Disabled Workers



APPENDIX C

GLOSSARY

GLOSSARY

CAN	Claim Account Number
CDR	Continuing Disability Review
CDRCF	Continuing Disability Review Control File
CER 100% Field File	– Characteristics Extract Record 100% Field File
DAC	Disabled Adult Child
DBAD	Disabled Beneficiary and Dependents Extract
DCF	Disability Control File
DDS	Disability Determination Services
DER	Detailed Earnings Record
DI	Social Security Disability Insurance
EN	Employment Network
EPE	Extended Period of Eligibility
EXR	Expedited Re-instatement Process
HI	Hospital Insurance
HUN	Housed Under Number
IRWE	Impairment Related Work Expense
LRF	Longitudinal Record Format
MBR	Master Beneficiary Record
MBR810 extract	– Master Beneficiary Record – version number
MBR814 extract	– Master Beneficiary Record – version number
MEF	Master Earnings File
Numident	Number Identification
OASDI	Old Age Survivors and Disability Insurance

OIM	Office of Information Management
PAN	Person's Account Number
PASS	Plan to Achieve Self-Sufficiency
PHUS	Payment History Update System
REMICS	Revised Management Information Counts System
RIB	Retirement Insurance Benefits
RSA	Rehabilitative Services Administration
SMI	Supplemental Medical Insurance
SAS	originally "Statistical Analysis System"
SER	Summary Earnings Record
SSA	Social Security Administration
SSI	Supplemental Security Income
SSI-LF	Supplemental Security Income - Longitudinal File
SSR	Supplemental Security Record
SVRA	State Vocational Rehabilitation Agency
T2	Title II (DI)
T16	Title XVI (SSI)
TRF	Ticket Research File
TTW	Ticket to Work
VR	Vocational Rehabilitation
YTD	Youth Transition Demonstration